NR INDEKSU 353965
PL ISSN 0860-1674

12

Z MIKROKOMPUTEREM NA TY

Bajtek

MIESIĘCZNY DODATEK DO SZTANDARU MŁODYCH NR 12 (24) GRUDZIEŃ 1987 CENA 100 ZŁ

w numerze: GRAFIKA KOMPUTEROWA



SM SZTANDAR MŁODYCH

* WYDANIE ŚWIĄTECZNE *

OPERACJA SUMOWANIA

Kluby komputerowe są najbardziej demokratyczną formą zapewniającą młodzieży dostęp do mikrokomputerów. Wiele osób ma już bowiem u siebie w domu te urządzenia, ale jeszcze więcej osób nigdy ich mieć nie będzie, są bowiem zwyczajnie za drogie jak na kieszeń większości polskich rodzin. Dlatego właśnie tak dużo uwagi poświęcaliśmy do tej pory i w dalszym ciągu poświęcać będziemy klubom komputerowym. Z każdego komputera będącego w dyspozycji klubu korzysta kilkanaście lub kilkadziesiąt osób. Dlatego tak ważne jest aby tych klubów, dobrze wyposażonych i fachowo prowadzonych, było jak najwięcej.

Ogłaszając rok temu Ogólnopolski Konkurs Klubów Komputerowych o Złotą Dyskietkę „Bajtka" chcieliśmy spopularyzować doświadczenia klubów najdłużej działających i dowiedzieć się jakie problemy mają, ci którzy dopiero zaczynają. Chcieliśmy stworzyć możliwość wymiany doświadczeń pomiędzy klubami, popularyzować pomysły najciekawsze i sprawdzone w praktyce. Chcieliśmy pomóc tym, którzy już „chcą chcieć" ale jeszcze nie wiedzą jak to się robi, chcieliśmy pomóc im zdobyć materiał do przemyśleń i wskazówki do praktycznego działania.

Oczekiwania te zostały w pełni spełnione. Publikowaliśmy już w poprzednim numerze „Bajtka" wyniki konkursu. Z jego zwycięzcami spotkaliśmy się na uroczystym podsumowaniu i wręczeniu nagród w dniu 2 grudnia 1986 roku. Była to chwila ważna dla wszystkich. Dzięki życzliwości przewodniczącego Komitetu ds. Młodzieży i Sportu, kierownictwa Związku Młodzieży Wiejskiej i związanej z nim „Agrotechniki", Rady Krajowej Turnieju Młodych Mistrzów Techniki oraz Centralnej Składnicy Harcerskiej wszyscy główni zwycięzcy konkursu wyjechali ze spotkania w warszawskim Stołecznym Domu Kultury Nauczyciela z nowymi komputerami. Złota Dyskietka (razem z Amstradem 6128) pojechała do Ostrowa Wielkopolskiego.

Ale spotkanie nasze było udane nie dlatego tylko, że część jego uczestników została obdarowana nagrodami. Co najmniej równie ważna była długa, koleżeńska rozmowa przy kawie i herbacie o problemach ruchu klubów komputerowych. Że są one potrzebne i należy je aktywnie rozwijać — nikt nie miał co do tego wątpliwości. Koledzy z Pałacu Młodzieży w Szczecinie opowiadali nam, że 1 września, w dniu zapisów do działającego tam klubu mikrokomputerowego, ustawiła się kolejka złożona z ponad 500 osób. Tymczasem na wszystkie zmiany można było maksymalnie przyjąć do pracowni komputerowej 80 osób! Reszta odeszła z kwitkiem. Miło nam swoją drogą, że głównym kryterium przesądzającym o tym kto zostanie przyjęty było wykazanie się czytaniem „Bajtka"!

Odwrotnie ma się sytuacja w klubie „Feniks" działającym przy Miejsko-Gminnym Ośrodku Kultury w Reczu w województwie gorzowskim. Jest to małe miasteczko i do klubu mogą przyjąć każdego chętnego. Ale brakuje im fachowych instruktorów. Wprawdzie dyrektor domu kultury jest informatykiem, ale będąc nawet wielkim entuzjastą mikrokomputerów nie może przecież poświęcić im całości swego czasu... Od razu podczas naszego spotkania pomoc instruktażową dla kolegów z Recza zaproponował Krzysztof Krupa, szef sieci klubów komputerowych TMMT. Przyświecająca nam przy organizowaniu konkursu idea wymiany doświadczeń pomiędzy różnymi klubami i instytucjami zainteresowanymi rozwojem informatyki popularnej potwierdziła się w ten sposób w całej rozciągłości.

Wszyscy z nieukrywaną zazdrością słuchaliśmy natomiast podczas warszawskiego spotkania opowieści kolegów z „Merizapu" w Ostrowie Wielkopolskim. Dzięki patronatowi miejscowych zakładów elektronicznych MERA-ZAP klub ten nie narzeka ani na brak sprzętu (wkrótce wzbogacą się o 20 IBM-ów!) ani na brak kadry fachowej. Zasygnalizowano natomiast brak odpowiedniej literatury niezbędnej dla właściwego korzystania z posiadanego sprzętu. Cała nadzieja w tym, że wiele takich pozycji zostało już przygotowanych do druku między innymi przez kluby komputerowe sieci TMMT i jeśli uda się wywalczyć odpowiedni przydział papieru — literatura ta zapewne będzie. Rzecz w tym, aby była jak najszybciej.

Doszliśmy wspólnie do wniosku, że kluby komputerowe stoją teraz przed nowym wielkim wyzwaniem. Bo o ile do tej pory głównym ich zadaniem było elementarne zapewnienie młodzieży dostępu do komputerów i nauka podstaw programowania, to obecnie należy uczynić kolejny krok — przejść do szerszych, praktycznych zastosowań komputerów. Pilnym zadaniem jest choćby stworzenie sieci informatycznych grupujących poszczególne rodzaje klubów... Dostatecznie długo komputer był w klubie jak gdyby zabawką samą w sobie; teraz należy doprowadzić do tego, aby był on praktycznie wykorzystywany...

To tylko niektóre z wątków poruszanych w dyskusji, jaka rozwinęła się po uroczystości wręczenia nagród. Będziemy do tych problemów stale na naszych łamach wracać. A już wkrótce ogłosimy kolejny II Ogólnopolski Konkurs Klubów Komputerowych o Złotą Dyskietkę „Bajtka"! Szczegóły w następnych numerach.

Waldemar Siwiński





W Nowym Roku życzymy wszystkim Czytelnikom „Bajtka" ciekawych programów, nowych rekordów w grach komputerowych i wszelkiej pomyślności. Redakcja.

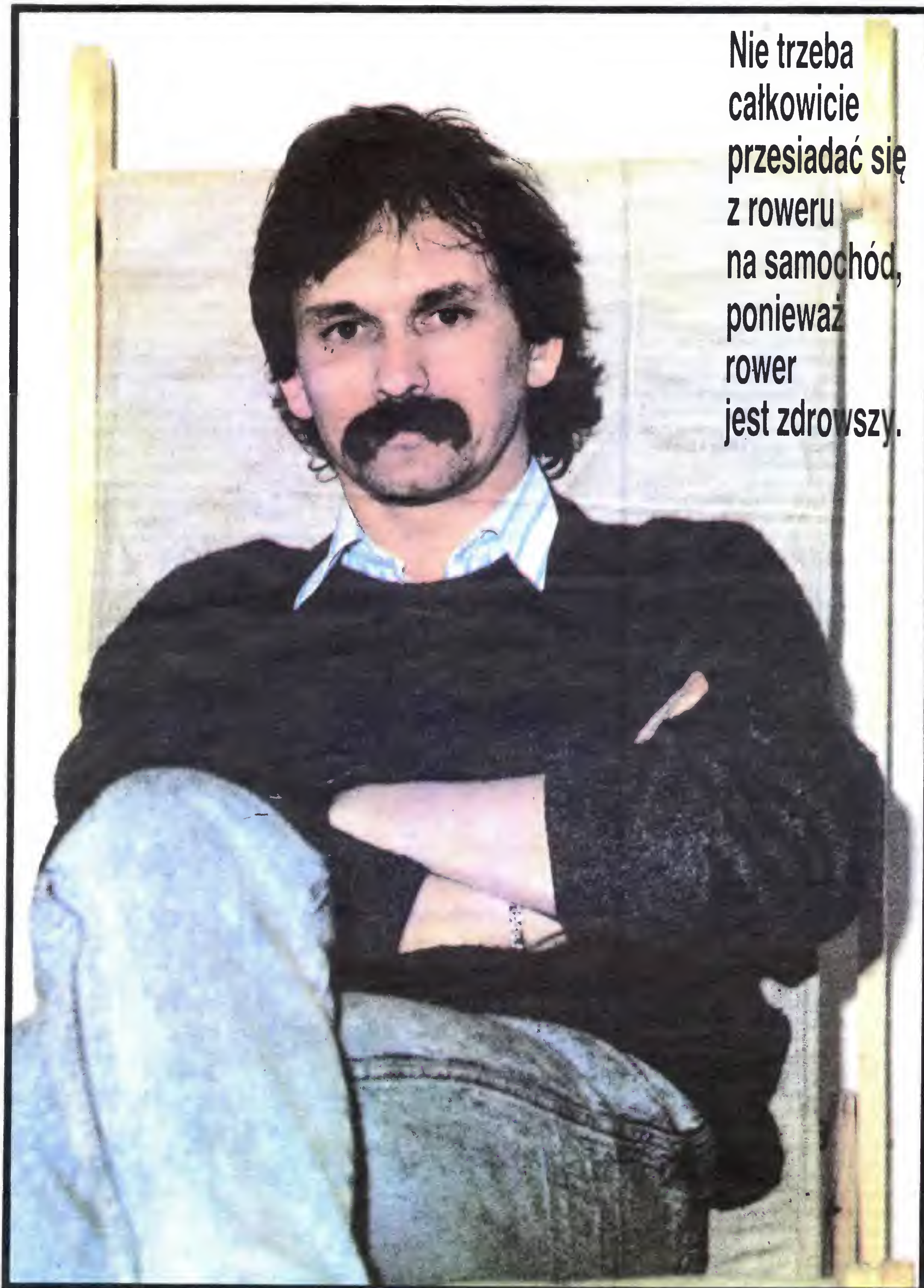
WYBIERZ SAM

„BAJTEK" — MIESIĘCZNY DODATEK DO „SZTANDARU MŁODYCH"
ADRES: 00-687 Warszawa, ul. Wspólna 61. Tel. 21-12-05
Przewodniczący Rady Redakcyjnej: Jerzy Domański-redaktor naczelny „Sztandaru Młodych".
ZESPÓŁ REDAKCYJNY: Waldemar Siwiński (z-ca redaktora naczelnego „SM" — kierownik zespołu „Bajtka"), Roman Poznański (z-ca sekretarza redakcji „SM" — sekretarz zespołu „Bajtka"), Krzysztof Czernek, Sławomir Gajda (red. techniczny), Andrzej Gogolewski, Andrzej Kowalewski, Andrzej Podulka, Sławomir Polak, Wanda Roszkowska (opr. graficzne), Kazimierz Treger, Marcin Waligórski, Roman Wojciechowski. Zdjęcia w numerze: Leopold Dzikowski.
Klany redagują:
Commodore — Klaudiusz Dybowski,
Amstrad-Schneider — Tomasz Pyć, Sergiusz Wolicki,
Spectrum — Marcin Przasnyski, Michał Szuniewicz,
Atari — Wiesław Migut, Wojciech Zientara.
Fotoskład — Tadeusz Olczak,
Montaż offsetowy — Grażyna Ostaszewska,
Korekta — Maria Krajewska, Ewa Mowińska.
WYDAWCA: RSW „Prasa-Książka-Ruch" Młodzieżowa Agencja Wydawnicza, al. Stanów Zjednoczonych 53, 04-028 Warszawa. Telefony: Centrala 13-20-40 do 49, Redakcja Reklamy 13-20-40 do 49 w. 403, 414.
Cena 100 zł.
Skład techniką CRT-200, przygotowalnia offsetowa i druk: PRASOWE ZAKŁADY GRAFICZNE RSW „PRASA-KSIĄŻKA-RUCH" w Ciechanowie, ul. Sienkiewicza 51
Nr zlecenia 138827 nakład 250 000 egz , K-109



Bajtek

TYLKO NARZĘDZIE


Nie trzeba całkowicie przesiadać się z roweru na samochód, ponieważ rower jest zdrowszy.

Rozmowa z Andrzejem Pągowskim, grafikiem, inicjatorem powstania Studia EGA — grupy plastyków posługujących się w pracy komputerami.

— Jak się zaczęła Twoja przygoda z komputerem?

— W listopadzie zeszłego roku nawiązałem współpracę z firmą komputerową REFLEKS. Zajmowałem się reklamą firmy. Komputery napawały mnie na początku szalonym obrzydzeniem. Niemal ze współczuciem patrzyłem na ludzi siedzących przed ekranami i wpatrzonych w rzędy cyferek, tabelki, niezrozumiałe napisy.

W pewnym momencie pojawiła się karta EGA do IBM-a, która zwiększyła możliwości graficzne komputera w tworzeniu grafiki. Zobaczyłem, że przy pomocy tego urządzenia mogę rysować na ekranie. Nastąpiło to co kiedyś, gdy byłem dzieckiem i tata dał mi do ręki ołówek i czystą kartkę papieru. Biel monitora zafascynowała mnie w tym samym stopniu co biel czystej kartki.

Ponieważ nie miałem czasu, by usiąść przed komputerem i uczyć się, wspólnie z szefem REFLEKSU wpadliśmy na pomysł zorganizowania pleneru plastyczno-komputerowego. Pierwszy plener odbył się w maju tego roku, drugi we wrześniu. Nie miałem wtedy pojęcia, że powstanie z tego jakaś grupa twórcza. Bardziej zależało mi na tym, by zaprosić kilku kolegów i pokazać im możliwości, które sam niedawno odkryłem. Efektem drugiego pleneru było utworzenie Studia EGA, które REFLEKS wyposażył w sprzęt. Nazwa pochodzi oczywiście od karty graficznej do IBM.

Do Studia należą (w kolejności alfabetycznej): Witold Dybowski, Maciej Kałkus, Andrzej Pągowski, Witold Popiel i Mieczysław Wasilewski.

— Komputer w plastyce to dla Ciebie nowa jakość, czy tylko kolejne narzędzie pracy?

— Komputer jest tylko narzędziem, ale żadne z narzędzi, których używałem dotychczas, nie narzucało takich zmian w sposobie projektowania. Na papierze postawienie każdej kreski jest nieodwracalne. Jeśli postawię ją źle, nie mam możliwości wycofania się. Pracując z komputerem mogę wszystkie swoje kolejne gesty zapisywać w pamięci, śledzić i korygować proces powstawania obrazu. W ciągu jednej sekundy mogę sprawdzić, czy rysunek jest wykonany dobrą grubością linii, itd.

Inna sprawa to liternictwo. Przyznam się, że nie jest ono moją najmocniejszą stroną. Zresztą 90 procent grafików w Polsce ma te same kłopoty. Komputer pozwala korzystać z wielu krojów pisma, co daje możliwość dokonywania prób i wyboru optymalnych zestawień.

Pozwala także na stuprocentową powtarzalność elementów graficznych. Można w ten sposób wykorzystywać wielokrotnie identyczny motyw bez konieczności np. fotografowania i naklejania elementów. Można wprowadzać do pamięci komputera nie tylko sam rysunek, ale także jego powiększenia, pomniejszenia, przeróżne deformacje. Przypomina to nieco zabawę klockami i nie jest możliwe do osiągnięcia — przynajmniej w takim stopniu — w żadnej innej technice.

Przy tym spełniło się jeszcze jedno moje skryte marzenie,... ja bardzo nie lubię się brudzić, a komputer daje możliwości pracy w warunkach wręcz sterylnych.

— Czy istnieją dziedziny lub tematy w sztuce, do realizacji których komputer nie może być zastosowany?

— To nie takie proste. Ostatnio zrobiłem na komputerze plakat do sztuki

GRA O JUTRO



Szekspira. Decyzja zależy wyłącznie od autora i tego, jaki chce przekazać klimat.

— Czy Twój plakat do filmu „Bal" wykonany na komputerze byłby równie finezyjny i nastrojowy?

— Korzystam z komputera tylko do robienia tych rzeczy, które same sobie na to pozwalają. Użycie komputera do realizacji takiego tematu jak „Bal" byłoby przestępstwem.

Niektórzy zarzucają mi, że często zmieniam techniki i maniery. Uważam, że tak właśnie trzeba. Technikę należy dostosować do tematu.

— Komputer — oczywiście odpowiednio zaprogramowany — potrafi reagować odpowiednio do zachowania się człowieka. Czy widzisz w tym szansę na powstanie „żywej plastyki", mogącej dopasowywać się do nastroju i wrażliwości odbiorcy, niejako współkreowanej przez niego?

— Myślę, że należy być ostrożnym w kontaktach ze wszystkimi nowymi technikami. Przede wszystkim nie należy popadać w nieuzasadnioną fascynację. Nie trzeba całkowicie przesiadać się z roweru na samochód ponieważ... rower jest zdrowszy. Komputer może robić prawie wszystko. Zależy to od sprzętu, talentu programisty itd. Ale czy to jest takie zabawne i czy ten sposób sprawdzania wrażliwości byłby dobry? Myślę, że chyba nie.

Bardzo chętnie słucham muzyki — ostatnio z płyt kompaktowych — i niestety, słuchając nagrań najwyższej jakości nieczęsto zdarza mi się, mieć odczucia takie, jak na koncercie w filharmonii.

— Nie istnieją narzędzia doskonałe. Dotyczy to z pewnością także komputerów. Jakie ograniczenia wynikają z ich stosowania w Twojej pracy?

— Praca z komputerem to coś takiego, jakby się z „malucha" przesiąść do „mercedesa". Wydaje nam się, że potrafimy prowadzić samochód, a jednak musimy się wszystkiego uczyć od nowa.

Ale poważnie. Nam brakuje rzeczy, które już istnieją. Korzystamy z dość prostego sprzętu i oprogramowania. Są monitory o olbrzymiej rozdzielczości, specjalistyczne komputery do tworzenia grafiki, o znacznie większych niż nasze IBM-y prędkościach przetwarzania danych. Nie łudźmy się jednak, że na Zachodzie jest to sprzęt dostępny dla każdego grafika. Tam też jest to szalenie drogie i tylko najbogatsze firmy stać na takie wyposażenie.

Największy kłopot mamy jednak z utrwalaniem naszej twórczości na papierze. Wydruki z kolorowych drukarek, rysunki z ploterów, zdjęcia z ekranu monitora, to wszystko ciągle jeszcze nie zapewnia wystarczającej palety barw. Ja stosuję paletę czarno-białą, w której kolory zastępowane są różnymi rastrami. Dzięki komputerowi mogę jakby za jednym pociągnięciem pędzla wypełnić wybraną przestrzeń dowolnym deseniem. Wszystkie moje obecne prace wykonuję techniką mieszaną. Na czarno-biały wydruk z komputera nakładam kolory w sposób tradycyjny.

— Nie masz więc zamiaru zrezygnować z farb, pędzla i palety?.

— Jest to niemożliwe. Nawet siadając przed komputerem mam obok ołówek i kartkę, na której robię sobie pierwszy szkic. Już samo poruszanie się po ekranie jest dość skomplikowane — myszką rysuje się zupełnie inaczej niż ołówkiem.

— Techniki komputerowe dają możliwość transformowania rzeczywistych obrazów, takich jak fotografia czy film, w celu stworzenia nowej jakości. Można więc sobie wyobrazić, że wiekopomne dzieła powstawać będą przez naciśnięcie kilku klawiszy. Czy jest to wizja realna?

— Trudno powiedzieć, co jest dziełem sztuki. Kreska postawiona przez uznanego artystę urasta do miana dzieła sztuki. Ta sama kreska postawiona przez kogoś innego pozostaje zwykłą kreską. Dzieła sztuki nie można więc oceniać mierząc czas jego powstawania. Liczy się to, czy potrafi ono wywołać reakcję w odbiorcy, poruszyć go. A czy będzie to pejzaż, czy maleńki kwadrat na białym tle, jest to rzecz drugorzędna.

— Ludzie pióra otrzymali komputerowe narzędzia — edytory tekstu, które poprawiają ich błędy ortograficzne, a nawet stylistyczne. Czy wyobrażasz sobie program graficzny korygujący np. błędy kompozycji obrazu?

— W tej dziedzinie nie jest to możliwe. Brak jest bowiem ściśle określonych zasad, takich choćby jak w ortografii. W malarstwie, plastyce dominującą rolę odgrywa intuicja, typowo ludzkie poczucie estetyki, a tego komputer nie jest w stanie się nauczyć. Natomiast dzięki komputerowi mogę sam siebie poprawiać znacznie łatwiej.

— Czy znasz się na komputerach? To znaczy, czy wiesz „co tam jest w środku" i czy potrafisz programować?

— Nie znam się na tym i — powiem więcej — nie chcę się znać. Po prostu nie jest mi to do niczego potrzebne. Komputer traktuję tak samo jak samochód, jeśli coś nie gra — jadę do mechanika. W naszej pracowni mamy profesjonalnych informatyków, którzy w każdej chwili służą nam wszelką pomocą. Nie muszę więc zastanawiać się „co tam siedzi w środku".

— Czy oznacza to, że niedługo wszyscy plastycy będą korzystali z techniki komputerowej?

— Jest to niemożliwe choćby z jednego powodu — nasze państwo nie jest w stanie sponsorować wszystkim grafikom. Jedyna szansa w firmach takich jak REFLEKS. Myślę, że czas wrócić do mecenatu w najlepszym tego słowa znaczeniu. W wielu krajach mecenat sztuki prowadzą firmy zajmujące się na codzień zupełnie czymś innym. Nie jest to działalność całkowicie bezinteresowna, a raczej lokata kapitału. Okazuje się, że na wspomaganiu artystów można nieźle zarobić.

Rozmawiali:
Wanda Roszkowska
Roman Poznański

SWEGO NIE ZNACIE

DIAGNOSTA Z DYSKIETKĄ!

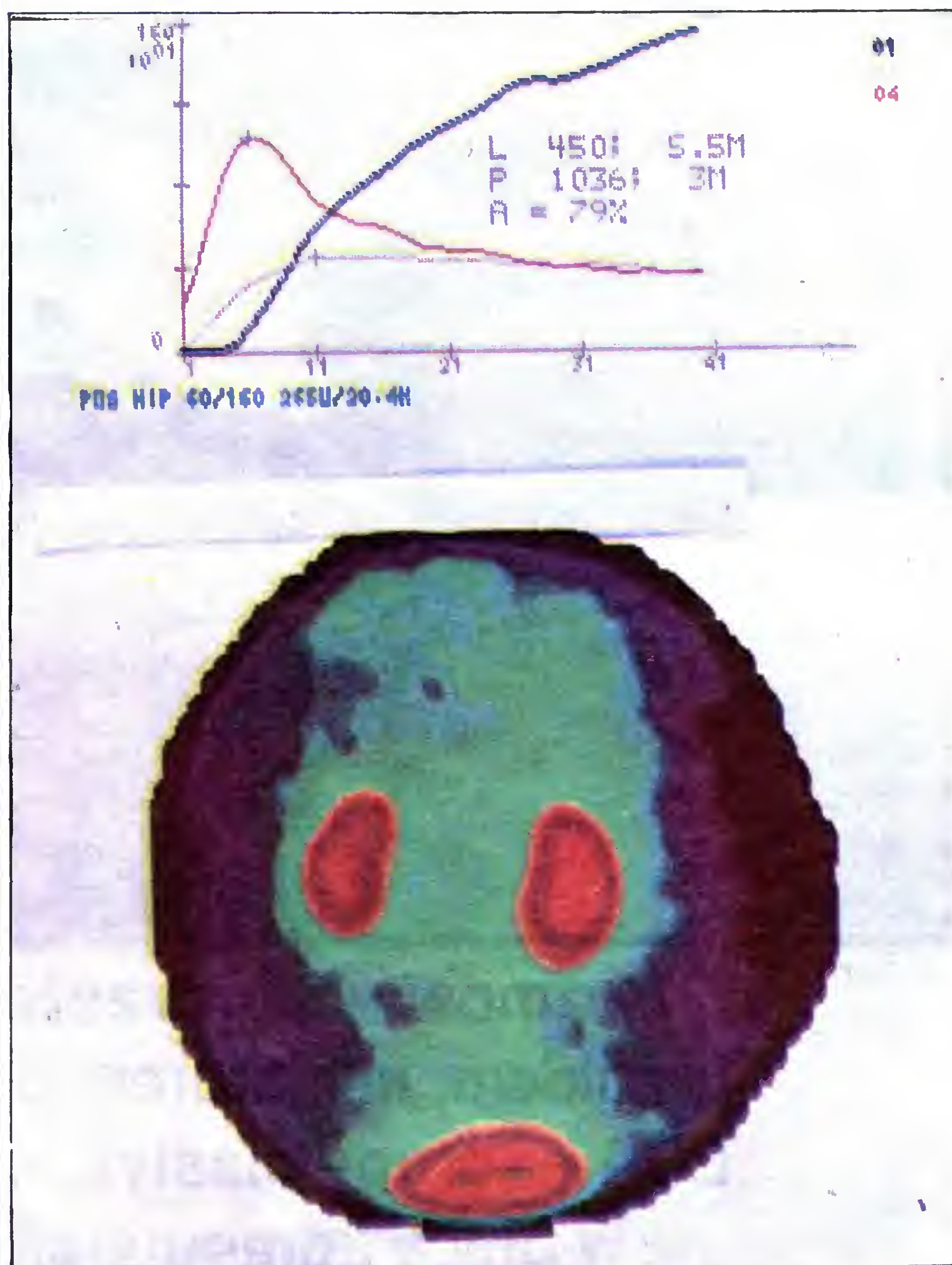
Kolorowy obrazek jest dokładny i przejrzysty. Oto prawa nerka — widać, że pracuje normalnie. Z lewą nie wszystko jest w porządku. Koniecznie trzeba poddać się kuracji.

Obrazki takie jak powyższy nie są oczywiście rejestracją na kliszy rentgenowskiego zdjęcia. Widać je na monitorze ustawionym w jednej z sal Szpitala Klinicznego Akademii Medycznej przy ul. Banacha w Warszawie, minikomputera firmy Sopha Medical. W jaki sposób tam trafiły? Co właściwie i jak można z nich odczytać?

Promieniowanie gamma nie należy do rzeczy najlepiej tolerowanych przez nasze organizmy. W dużych dawkach bywa groźne, czasem śmiertelne. Nawet zwykłe badanie rentgenowskie nie jest obojętne dla organizmu, wszak w jego trakcie nasze ciało na wskroś przenikają „niesympatyczne" promienie. A diagnostyka medyczna wymyśliła coś jeszcze — scyntygrafię. Zamiast prześwietlać pacjenta promieniami, lepiej w pewnych przypadkach, by sam nasz organizm zaczął je emitować. Oczywiście sam tego nie zrobi, promieniowanie wywołane będzie przez izotop, który wniknął do jego wnętrza.

Od strony teorii sprawa wydaje się prosta. Pacjentowi podaje się doustnie, dożylnie lub poprzez wdychanie substancję chętnie i szybko przyswajaną przez organizm. Substancja ta skażona jest promieniotwórczym izotopem. Izotop ten dociera do celu a urządzenie zwane gammakamerą śledzi promieniowanie. W ten sposób można dowiedzieć się wiele, zarówno o samym procesie transportu — np. czy nerki pacjenta są drożne, jak długo zalega w nich izotop itd., a także — w przypadku wspomnianych absorbowanych przez dane organy substancji — przeprowadzić badania statyczne tych organów.

Teoria jest prosta, praktyka znacznie bardziej złożona. Z jednej strony bowiem ze względu na szkodliwość promieniowania podawać trzeba bardzo małe dawki izotopu, z drugiej starać się, by miał on jak najkrótszy tzw. czas połowicznego rozpadu, czyli innymi słowy, by jak najkrócej oddziaływał na nas promieniami gamma. Impulsy jakie odbiera gammakamera są zatem dość słabe, a przecież dla dokładności badania trzeba oddzielić je od



tła i wyeliminować zakłócenia jakie przynosi m.in. efekt Comptona (kłania się fizyka!). Dlatego m.in. nie wystarcza gammakamera, klisza fotograficzna i lekarz. Potrzebny jest jeszcze komputer, a czasem także programista.

Programy, które wykorzystuje komputer „zatrudniony" w diagnostyce nuklearnej — wyjaśnia Andrzej Spychała zatrudniony w klinice na Banacha fizyk, do niedawna jeszcze nauczyciel w popularnym „Czackim" — mają specyficzną budowę. Składają się one z ciągu podprogramów. Język jakim taki makroprogram jest napisany charakteryzuje dany system. Poszczególne jego komendy — procedury umożliwiające np. odcięcie tła, czy filtrację zakłóceń są oczywiście „rozbieralne". Można również wedrzeć się do poszczególnych procedur i zmieniać je z poziomu języków programowania — Fortranu, czy Assemblera. Dla standardowych badań i ich obróbki komputerowej informatyk nie jest więc potrzebny, ale już pisanie nowych makroprogramów to zadanie przekraczające z pewnością możliwości technika czy lekarza.

Komputery używane do obróbki danych w scyntygrafii nie różnią się specjalnie od „zwykłych" PC. Wspomniane już ograniczenia fizyczne i medyczne decydujące o tym, że w zasadzie nie można dziś myśleć o podawaniu izotopu dającego promieniowanie o większym natężeniu mają wpływ na wymagania stawiane komputerowi. Nie są one zbyt wielkie. Do obsługi badań wystarczy ok. 128 KB pamięci operacyjnej. Natomiast konieczność sprawnej obróbki obrazu sprawia, że stojące na Banacha minikomputery wyposażone są w koprocesory obrazowe. Cały system oparty jest na bazie PDP11 i korzysta z systemu operacyjnego RT 11.

Przyglądamy się obróbce danych. Na stole diagnostycznym nie ma pacjenta, obraz z gammakamery został już kilka dni temu zapisany na dysku o pojemności 140 MB. Wywołujemy z klawiatury dane pacjenta i na ekranie monitora pojawiają się kolejne obrazki zarejestrowane podczas badania nerek. Na pierwszych z nich czerwono zaznaczony izotop dociera dopiero do nerek pacjenta, na kolejnych pojawia się w nerkach, by w końcu zgromadzić się w pęcherzu moczowym. Widać jednak wyraźnie nieprawidłowość pracy jednej nerki, izotop dostawszy się tam nie idzie dalej, nerka nie działa, a w każdym razie pracuje dużo wolniej od drugiej. Obraz można uśrednić w czasie, z czterdziestu kilku otrzymujemy jeden — widoczny na naszej ilustracji. Komputer ma także możliwość wybrania z obrazu, za pomocą drążka sterowego lub z klawiatury tzw. ROI, czyli obszarów zainteresowania. W ten sposób na kolejnym obrazku ukazują się już tylko te narządy lub ich fragmenty, które interesują diagnostę. Obraz można analizować na monitorze lub drukować. My, dla potrzeb „Bajtka" skorzystaliśmy z tej drugiej możliwości. Nasza poligraficzna niedoskonałość nie pozwala na obiektywną ocenę pracy drukarki, ale, uwierzcie nam, jest ona bez zarzutu.

Scyntygrafia — mówi Andrzej Spychała — nie zastępuje innych badań diagnostycznych, lecz je uzupełnia. Badamy zresztą nie tylko pracę poszczególnych narządów, wykonywana jest także scyntygrafia całego ciała (w tym układu kostnego). Badanie to byłoby jeszcze dokładniejsze gdybyśmy mogli posługiwać się izotopem dającym promieniowanie o większym natężeniu. Oczywiście z racji na szkodliwą działalność promieniowania gamma należałoby zapewnić krótszy czas jego emisji, czyli okres połowicznego rozpadu izotopu — kilka minut zamiast kilku godzin. Wowczas jednak izotopy musiałyby być wytwarzane bezpośrednio w samym szpitalu.

Diagnostyka komputerowa nie jest obca naszym informatykom. Np. w Instytucie Radioelektroniki Politechniki Warszawskiej powstał oryginalny system operacyjny dla medycyny nuklearnej pod nazwą „Gamma-PW". Jego twórcy z doc. Romanem Szabatinem na czele oferują go nie tylko polskim placówkom służby zdrowia.

Chociaż wyposażenie naszych klinik i szpitali w sprzęt wysokiej klasy nie jest, delikatnie mówiąc, najlepsze, scyntygrafię komputerową stosuje się już w wielu placówkach. Czekamy natomiast na uruchomienie pierwszych w Polsce pracowni diagnostycznych medycyny nuklearnej nowej generacji opartej na wykorzystaniu tzw. jądrowego rezonansu magnetycznego (NMR). Metoda ta dająca m.in. znakomitą wręcz zdolność rozdzielczą obrazu to nowy krok w poznaniu tajemnic ludzkiego ciała. To także nowe zadania dla sojuszników lekarzy — komputerów i ich programistów. Komputery poradzą sobie z pewnością.

Krzysztof Nowostawski
Grzegorz Onichimowski

PROGRAMOWAĆ MOŻE KAŻDY

KAMERA, TELEWIZOR I... REKURENCJA

Jeśli kamerę, do której podłączony jest telewizor skierujemy na ekran tegoż telewizora, uzyskamy ciekawy efekt. Na ekranie pojawi się telewizor, na którego ekranie pojawi się telewizor, na którego ekranie... itd.

Opisane zjawisko stało się pretekstem do napisania całkowicie „nieużytkowego" (ale za to śmiesznego) programu w LOGO. I chociaż ten program jest typowym przykładem „sztuki dla sztuki" może służyć jako doskonały model zastosowania rekurencji, którą w tym przypadku posłużymy się do uzyskania kolejnych, zmniejszających się obrazów

```
to obrazy
telewizor :krok
make "krok :krok * 2 / 3
obrazy
end
```

I to jest właśnie rekurencja. W programowaniu polega ona na korzystaniu podczas definiowania procedury z tejże definiowanej procedury. O tym, jak potężnym narzędziem jest rekurencja świadczy fakt, że króciutka procedura **obrazy** realizuje praktycznie w całości program „kamera-telewizor". Wystarczy nadać wartości początkowe zmiennej **krok** i kamera w ruch.

```
to kamera
fs cs
make "krok 24
obrazy
end
```

Na tym właściwie powinienem skończyć. Bo przecież każdy wie jak wygląda telewizor i sam może go sobie zaprojektować. Roztargnionym przypominam jednak co składa się na odbiornik TV.

```
to telewizor :krok
skrzynka :krok
ekran :krok
lewanoga :krok
prawanoga :krok
glosnik :krok
klawisze :krok
klapka :krok
suwaki :krok
end
```

Podczas rysowania poszczególnych elementów przyda nam się procedura ustawiająca żółwia w żądanym punkcie

```
to ustaw :x :y
pu home
rt 90 fd :x
lt 90 fd :y
pd
end
```

i rysująca prostokąt

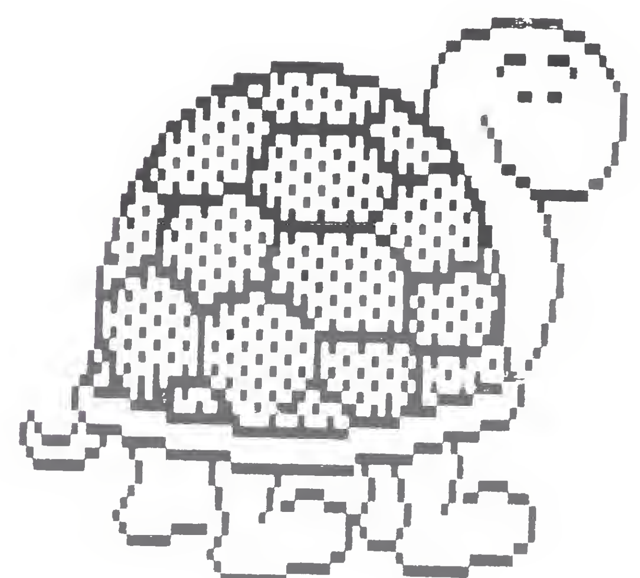
```
to prostokat :bok1 :bok2
repeat 2 [lt 90 fd :bok1 lt 90 fd :bok2]
end
```

Teraz możemy spokojnie przystąpić do projektowania elementów składowych odbiornika.

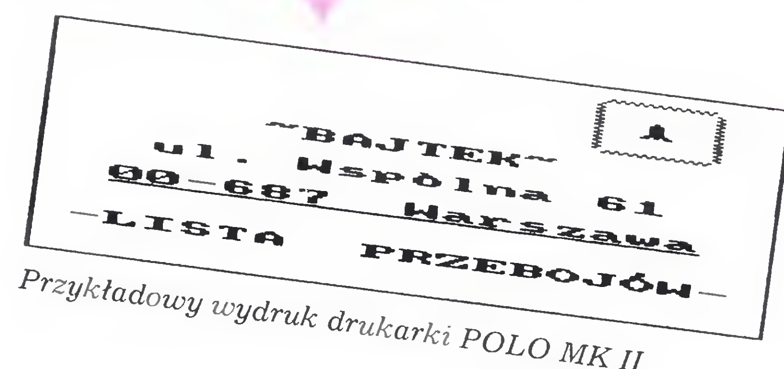
```
to glosnik :krok
ustaw :krok * 12 :krok * 6
lt 90
repeat 3 [lt 90 fd :krok * 4 rt 90 pu fd
 :krok / 2 pd rt 90 fd :krok * 4 lt 90
pu fd :krok / 2 pd]
end
to suwak :krok
lt 90
fd :krok / 2
rt 90 fd :krok / 3
prostokat :krok / 2 :krok * 2 / 3
lt 90 fd :krok / 2 lt 90 fd :krok / 3 rt
 90
fd :krok * 3 / 2
end
to klawisze :krok
ustaw :krok * 12 :krok * 1
repeat 5 [prostokat :krok / 2 :krok / 2
lt 90 fd :krok / 2 rt 90]
end
to klapka :krok
ustaw :krok * 12 0
prostokat :krok * 5 / 2 :krok
end
to suwaki :krok
ustaw :krok * 12 -:krok * 2 suwak :krok
ustaw :krok * 12 -:krok * 3 suwak :krok
ustaw :krok * 12 -:krok * 4 suwak :krok
ustaw :krok * 12 -:krok * 5 suwak :krok
end
to lewanoga :krok
ustaw -:krok * 7 -:krok * 7
prostokat :krok :krok
end
to prawanoga :krok
ustaw :krok * 11 -:krok * 7
prostokat :krok :krok
end
to skrzynka :krok
ustaw :krok * 13 :krok * 7
prostokat :krok * 23 :krok * 14
end
to ekran :krok
ustaw :krok * 9 :krok * 6
prostokat :krok * 18 :krok * 12
end
```

Dla tych, którzy po uruchomieniu programu mieliby ochotę poczekać aż skończy się on wykonywać, mam smutną wiadomość — ten program nigdy się nie skończy. Pracowity żółw będzie rysował wciąż mniejsze i mniejsze telewizory, nieświadom, że od pewnego momentu zacznie się uwijać w obrębie jednego punktu na ekranie.

Roman Poznański



Test przeprowadzono na drukarce dostarczonej przez brytyjską firmę wysyłkową ELECTRONICS EXPORT Londyn W5, P.O. Box 869, tel. z Polski 0-0441 993 7000. Serdecznie dziękujemy.



Przykładowy wydruk drukarki POLO MK II



TRITON

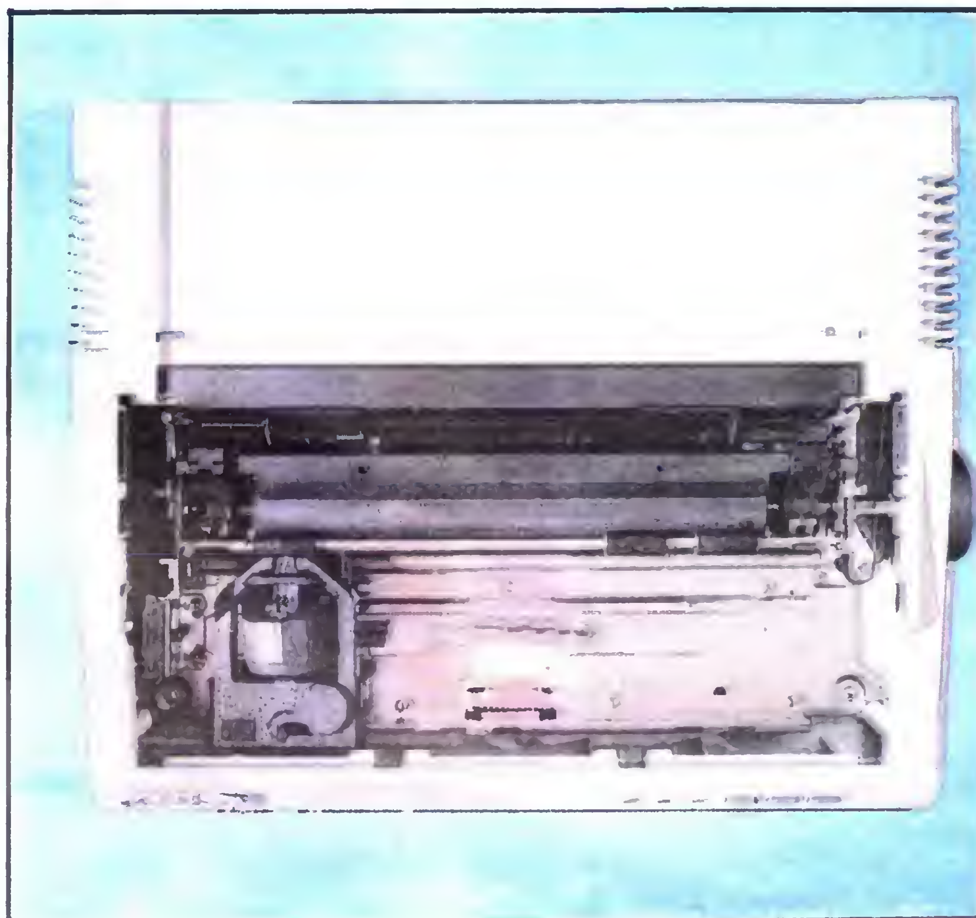
Drukarki komputerowe przypominają samochody: bywają laserowe Mercedesy i Mikrusy lub Trabanty (np. Seikosha GP-50). Tak więc testowaną przed miesiącem drukarkę STAR NX-15 można porównać z Volkswagenem Passatem, a na przykład polską D-100 — z Syrenką.

Obecnie testowaną drukarkę wyprodukowała nieznana w Polsce firma Radofin Elektronix z Hongkongu. Należy ona do serii drukarek o wspólnej nazwie Triton, różniących się między sobą szybkością druku i (jakże by inaczej) ceną.
Testowany egzemplarz nie jest właściwie drukarką Triton — jest on starszym, nieco uproszczonym modelem o nazwie Polo MK II, jednakże zgodnym programowo i pod względem parametrów użytkowych bardzo podobnym do reszty serii. Pod względem ceny należy do klasy drukarek popularnych, a kody sterujące odpowiadają standardowi Epson FX-80.

W porównaniu z testowaną uprzednio drukarką STAR NX-15, POLO sprawia wrażenie zabawki. Zawdzięcza je lekkiej i zwartej konstrukcji o opływowych, estetycznych kształtach. Mechanizm transportu papieru umożliwia używanie zarówno pojedynczych arkuszy, jak i perforowanych wstęg papieru o szerokości do 10 cali. Zakładanie pojedynczych arkuszy jest niestety bardzo uciążliwe i czasochłonne — brak spotykanego w wielu drukarkach automatycznego wciągania papieru.

Przełączniki sterujące na czołowej stronie drukarki umożliwiają wysuw papieru o linię bądź o stronę i programowe odłączenie drukarki od komputera (w drukarkach Triton — również wybór pisma zwykłego lub NLQ).

Dostęp do wnętrza drukarki jest utrudniony. Nawet przejrzysta pokrywa głowicy otwiera się dość ciężko, a otwarcie dodatkowej klapki umożliwiającej m.in. wymianę kasety z taśmą barwiącą wymaga wyrafinowanej mieszanki siły i sprytu nieszczęsnego użytkownika.
Jeszcze gorzej jest z dostępem do mikroprzełączników umożliwiających ustawienie standardowych parametrów drukarki podczas inicjacji. Wymaga to wykręcenia czterech śrub i zdjęcia górnej połowy obudowy drukarki, a co za tym idzie naraża na uszkodzenie wrażliwe układy scalone procesora i pamięci operacyjnej. Nowsze modele drukarek Triton wyposażone są w niewielką pokrywę usuwającą ten problem.


Widoczna głowica z taśmą barwiącą oraz elementy mechaniczne.

Dziesięciogłowa głowica przesuwana jest po metalowej szynie przy pomocy metalowej linki. Otaczający głowicę gumowy kołnierz zmniejsza hałas towarzyszący drukowi. Odstęp głowicy od papieru regulowany jest dwupozycyjnym przełącznikiem (instrukcja obsługi mówi o czterech pozycjach — prawdopodobnie dotyczy to nowych wersji drukarki). Siła uderzenia igieł o papier wystarcza do uzyskania 2 wyraźnych kopii drukowanego tekstu. Jakość i asortyment krojów pisma zdecydowanie wykracza poza wymagania stawiane drukarkom popularnym. Czytelność i estetyka liter w trybie zarówno zwykłym (DRAFT)., jak i korespondencyjnym (NLQ), dorównuje drukarkom o klasę droższym. Pismo może być zagęszczone lub podwójnej szerokości, podkreślone, proste lub pochylone, zwykłe lub korespondencyjne, tłuste lub podwójnie uderzane oraz w prawie dowolnych kombinacjach powyższych krojów. Szybkość druku wynosi około 100 znaków na sekundę w trybie zwykłym i 25 znaków na sek. w trybie korespondencyjnym (dane katalogowe dla drukarek Triton są nieco lepsze w zależności od typu). Praca drukarki nie jest nadmiernie głośna, a szczelna pokrywa dobrze zabezpiecza wnętrze przed kurzem.

Również asortyment trybów graficznych (siedem o gęstościach od 480 do 1920 punktów na cal) odpowiada raczej drukarkom średniej, a nie popularnej klasy. Widoczny jest co prawda ślad każdego przejścia głowicy przez arkusz papieru, ale zdarza się to również drukarkom bez porównania droższym. Natomiast miłym zaskoczeniem jest szybkość pracy podczas druku obrazu. Pod tym względem POLO wyraźnie przewyższa znacznie droższą drukarkę STAR NX-15, pomimo, a może właśnie dzięki mniej wyrafinowanej logice sterującej ruchem głowicy.

Dla polskiego użytkownika drukarek szczególnie istotna jest możliwość definiowania własnych znaków. POLO posiada tę możliwość, pod warunkiem wyposażenia w pamięć RAM o pojemności 8 KB. Przy ewentualnym zakupie należy zwrócić na to uwagę, albowiem istnieje również wersja z pamięcią 2 KB. Definiowanie dodatkowych znaków jest możliwe jedynie w przypadku, gdy komputer wyposażony jest w interface ośmiobitowy. Niestety, nie jest możliwe definiowanie własnych znaków o jakości korespondencyjnej (NLQ).

Drukarka okazała się niezastąpiona przy współpracy z ATARI 800 XL co Czytelnicy mogą sprawdzić w Klanie Atari niemniej również podczas współpracy z innymi komputerami spisywała się bez zarzutu.

W wyniku intensywnej pracy uległa nieodwracalnemu uszkodzeniu taśmą barwiącą w kasecie. Niestety jest to raczej nietypowa taśma o szerokości zaledwie 6 milimetrów i zmusiło to redakcję do sprowadzenia nowej kasety od producenta.
sprowadzenia nowej kasety od producenta.

Przykład grafiki

PODSUMOWANIE

Testowana drukarka POLO MK II (Triton) jest średniej klasy drukarką o korzystnym stosunku jakości do ceny. Cechuje się dobrej jakości drukiem w trybie zwykłym i korespondencyjnym oraz dużą szybkością pracy w trybie graficznym.

Zalety drukarki:
- małe wymiary i lekkość obudowy
- stosunkowo niewielka hałaśliwość pracy
- wysoka jakość druku w trybie zwykłym i korespondencyjnym
- duża liczba krojów i formatów liter
- możliwość definiowania polskich znaków
- duża pojemność bufora przyspieszająca współpracę z komputerem (w wersji z 8 KB RAM, z tego ok 5 KB jest wykorzystane przez bufor).

Wady drukarki:
- utrudniony dostęp do głowicy drukującej i przełączników sterujących na płycie montażowej
- brak mechanizmu ułatwiającego wciąganie pojedynczych arkuszy papieru
- niezbyt wysoka trwałość taśmy barwiącej w kasecie.

Tomasz Nowicki
Sławomir Polak

Parametry techniczne drukarek Triton w/g specyfikacji producenta:

- **głowica 9-igłowa, możliwość wymiany przez użytkownika**
- **druk dwukierunkowy w trybie zwykłym i jednokierunkowy w trybie NLQ oraz graficznym**
- **szybkość druku:**
  w trybie zwykłym 120—160 zn/sek (w zależności od wersji)
  w trybie NLQ 25—30 zn/sek
- **bufor 1 lub 4 KB (zależnie od rozmiaru pamięci RAM)**
- **złącze równoległe typu Centronics (opcjonalnie szeregowe RS—232 C)**
- **zgodność programowa z drukarkami Epson FX—80 i IBM (z wyjątkiem grafiki)**
- **standardowo wmontowana zębatka do przesuwu papieru perforowanego 10 cali (opcjonalnie — wersje z regulacją 4—10 cali)**
- **waga 5 kg**
- **rozmiary 362x295x76 mm.**

IGŁĄ PO PAPIERZE

Drukarka jest jednym z podstawowych urządzeń współpracujących z komputerem. Bez niej trudno sobie wyobrazić realizację jakiegokolwiek poważniejszego zadania. Konstruowaniem drukarek zajmuje się wiele firm na świecie. Powstało kilkanaście ciekawych rozwiązań technicznych, wśród nich drukarki termiczne, strumieniowe, laserowe, rozetkowe, drukarki do druku barwnego. Zdecydowanie największą popularność zdobyły sobie jednak drukarki mozaikowe. Niska cena, duże możliwości graficzne i prostota obsługi sprawiły, że spotykamy je dziś prawie w każdym biurze.

Najważniejszym elementem drukarki mozaikowej jest głowica zawierająca zwykle 9 lub 24 cieniutkie stalowe igły. Wydruk powstaje na papierze na skutek uderzeń tych igieł poprzez taśmę barwiącą. Obraz widoczny na arkuszu składa się więc z dużej liczby malutkich plamek, które przypominają ułożoną z drobnych kamyczków mozaikę.

ZNAKI

Kształt wszystkich znaków zapisany jest na stałe w pamięci ROM, lub definiowany przez programistę i przesyłany do RAM-u drukarki. Informacja ta zapisywana jest w nieco innej postaci niż znaki wyświetlane na ekranie.

Matryca literek pojawiających się na monitorze jest zazwyczaj kwadratem złożonym z 64 pól. Punkty matrycy mogą być zapalone lub wygaszone. Obserwując je z większej odległości mamy złudzenie, że tworzą ciągły obraz. Do jednoznacznego opisania wyświetlanego znaku wystarcza 8 liczb odpowiadających 8 poziomym rzędom.

Podstawowa matryca wydruku w drukarce jest prostokątem o wysokości 9 i szerokości 6 punktów (w przypadku głowicy dziewięcioigłowej). Punkty mogą być drukowane nie tylko wewnątrz pól, ale i na ich styku. Całkowita szerokość znaku jest mniejsza, lecz mamy za to do dyspozycji 11 kolumn przesuniętych względem siebie o 1/2 elementarnej działki. Kształt znaku opisuje 11 bajtów odnoszących się do kolumn wydruku. Przy projektowaniu nowych symboli graficznych musimy zwrócić uwagę, by sąsiadujące ze sobą w poziomych rzędach punkty nie zachodziły na siebie. Prędkość przesuwu głowicy jest tak duża, że drukarka nie nadążyłaby ich drukować.

Każdy ze znaków ma wysokość tylko 8 punktów i może zajmować górną lub dolną część podstawowej matrycy 9*6. W związku z tym w jego druku uczestniczy tylko 8 górnych lub dolnych igieł głowicy. Informacja o położeniu znaku jest określana angielskim terminem descender i umieszczana w dodatkowym bajcie zwanym bajtem atrybutu.

Zwróćmy uwagę, że nie wszystkie litery alfabetu mają tę samą szerokość. Przekonamy się o tym porównując „w" z „l". Jeśli tylko zechcemy, to drukarka uwzględni różnice szerokości znaków. W bajcie atrybutu można zapisać mumer początkowej i końcowej kolumny wydruku, uzyskując w ten sposób węższe litery. Druk różnicujący szerokość liter nazywa się drukiem proporcjonalnym. Niestety, nie wszystkie drukarki posiadają tę możliwość.

KROJE DRUKU

Współczesne drukarki umożliwiają wybór kilkunastu rodzajów druku. W związku z tym redagowany tekst może być bardzo urozmaicony graficznie bez konieczności żmudnego definiowania nowych znaków. Użytkownik dysponuje zwykle trzema zestawami liter o różnej szerokości: wąskie 17 znaków/cal, elite 12 znaków/cal i pica 10 znaków/cal. Szerokość każdego z nich można zwiększyć dwukrotnie w trybie zwanym expanded (breitdruck).

Wśród specjalnych rodzajów druku znajduje się zwykle pismo pochyłe (kursywa), podkreślone, pismo indeksowe górne (np. wykładniki potęg), indeksowe dolne oraz pismo korespondencyjne o wysokiej jakości zwane NLQ (Near Letter Quality). Oprócz tego istnieje druk tłusty i wyrazisty. W obu przypadkach znaki są drukowane dwukrotnie, przy czym dla tłustego druku drugie uderzenie wypada trochę z boku, a dla wyrazistego poniżej.

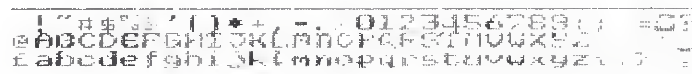
Drukarka umożliwia również regulację odstępu między wierszami, ustawienie długości strony, lewego, prawego, dolnego i górnego marginesu oraz tabulacji. Dla wygody użytkowników z różnych krajów przewiduje się również dostosowanie sprzętu do wymagań narodowych alfabetów. Na przykład drukarki STAR mają zakodowane w pamięci 8 zestawów znaków: amerykański, angielski, francuski, niemiecki, angielski, duński, szwedzki, włoski oraz hiszpański.

GRAFIKA

Spośród licznych możliwości drukarki najefektowniejsze jest niewątpliwie drukowanie całych obrazów. Służą do tego tryby graficzne różniące się między sobą gęstością i szybkością wydruku. Pozwalają one utworzyć dowolne rysunki. Gęstość druku waha się tu zwykle od 60 do 240 i więcej punktów na cal.

Zaprogramowanie wydruku jest procesem dość żmudnym mimo rozbudowanych i urozmaiconych funkcji realizowanych przez drukarkę. Przeciętny użytkownik nie będzie więc zajmował się tym zagadnieniem, lecz skorzysta z gotowego oprogramowania. Na rynku dostępna jest duża liczba edytorów tekstów, programów graficznych i specjalistycznych inżynierskich. Oprócz podstawowych operacji umożliwiających redagowanie tekstu, tworzenie wykresów lub diagramów dysponują one bogatym zestawem liternictwa rozszerzającego standardowe możliwości drukarki. Do programów graficznych dołączane są obrazki zapisane na dyskietkach, które można użyć do drukowania zaproszeń, wizytówek, ozdabiania listów.

Janusz Jarmoch





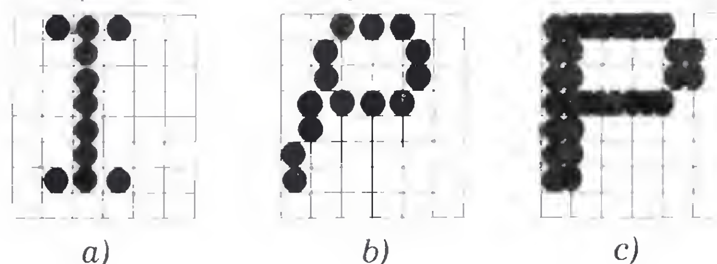
rys. 1 Dodatkowe zestawy liter z programu CHIWRITER

```
                         root (\)
        |                    |                        |
   filename 1          directory 1               directory 2
                 |           |           |         |            |
           filename 1   directory 3   filename 2  filename 1   filename 2
                 |           |           |
           filename 1   filename 2   filename 3
```

rys. 2 Przykład diagramu narysowanego przy pomocy programu CHIWRITER

$$y^{3\beta}_{\mathbf{n}_1 n_2}(p_1,p_2) = U^{3\beta}_{\mathbf{n}_1 n_1}(p_1,p_2) + \int_0^\infty \frac{dp_3\, p_3^2}{8\pi^3} \sum_{n} \sum_{\mathbf{n}} \sum_{\alpha_2} \sum_{\beta_2} \sum_{n'} \sum_{n''} (-1)^m \times \left[\frac{U^{33}_{\mathbf{n}_1 n_1}(p_1,p_2)}{p_3^2 - k^2} \right] z_{3\mathbf{n}_1 n_1} h_n(p_3,p_3)\, a^{\alpha_2 3}_{\mathbf{n}n(\mathbf{n}_1-\mathbf{n})n' n_2}\, a^{\beta_2\beta}_{\mathbf{n}n(-\mathbf{n}_1+\mathbf{n})n'' n_2}\, H^{\alpha_2\beta_2}_{(\mathbf{n}_1-\mathbf{n})n' n''}(p_3,p_2). \qquad (110)$$

rys. 3 Skomplikowany wzór matematyczny drukowany przez program CHIWRITER



a) b) c)

rys. 4 Różne rodzaje druku:
a) pica,
b) kursywa,
c) tłusty druk

POŻYCZONE LITERY

Wiele gier zadziwia nas swoim wymyślnym liternictwem. Czcionka pogrubiona, pochyła, „komputerowa", gotyk lub cyrylica, umieszczone na ekranie sprawiają, że strona graficzna programu zyskuje bardzo wiele w oczach użytkownika.

Jak zapewne wiadomo większości posiadaczy ZX SPECTRUM, cały zestaw znaków (96 liter) tego komputera można zastąpić nowym zestawem, zaprojektowanym przez użytkownika. Litery jakie widzimy na ekranie zaraz po włączeniu komputera, są pobierane z generatora znaków, który znajduje się pod adresem 15616 i zajmuje 768 bajtów (96 ∗ 8 bajtów). Adres pierwszej komórki generatora obliczany jest przez komputer na podstawie zawartości zmiennej systemowej CHARS (komórki o adresach 23606 i 23607), wg wzoru:

ADRES=(PEEK 23607)∗256+(PEEK 23606) +256

Zmieniając więc zawartość zmiennej, zmieniamy zarazem adres, spod którego pobierane są kształty znaków. Oczywiście nie wystarczy zmienić samego adresu, należy jeszcze pod nim umieścić dane o kształcie nowego zestawu liter. O tym, że projektowanie liter to długa i żmudna praca, nie trzeba chyba nikogo przekonywać i właśnie dlatego proponuję ułatwienie tego zadania przez wyciągnięcie interesującego kroju pisma z gotowych programów, aby móc go wykorzystać do własnych celów. Sposób jest prosty. Najpierw jednak należy wpisać i zapamiętać na taśmie zamieszczony obok program CZCIONKA.

— Do pamięci komputera wczytujemy za pomocą programu COPY-COPY segment gry, który jest „podejrzany" o to, że zawiera zapis kształtów liter (zwykle jest to najdłuższy segment);

— Po wczytaniu segmentu wychodzimy z COPY-COPY do BASIC-a (klawisz Y). Wczytana część ciągle pozostaje w pamięci.

— Wczytujemy z magnetofonu i uruchamiamy program CZCIONKA.

Na górze ekranu pojawi się okienko pokazujące zestaw liter z aktualnego generatora znaków (na który wskazuje zmienna CHARS). Prawie zawsze będą to jakieś przypadkowe kształty. Oto opis obsługi programu:

klawisz 1 — zmniejsza zawartość komórki 23607 o 1 (czyli adres generatora znaków zmniejsza się o 256)
klawisz 2 — zwiększa zawartość komórki 23607 o 1
klawisz 3 — zmniejsza zawartość komórki 23606 o 8 (adres generatora zmniejsza się o 8)
klawisz 4 — zwiększa zawartość komórki 23606 o 8
klawisz 8 — zmienia działanie klawiszy 3 i 4 w ten sposób, że zmniejszają lub zwiększają one adres generatora o 1. Ponowne naciśnięcie klawisza przywraca stan poprzedni.
klawisz 9 — uruchamia „automatyczne przeszukiwanie", tzn. program działa tak, jakby cały czas wciśnięty był klawisz 2. Ponowne wciśnięcie przywraca stan normalny.
klawisz 0 — wyjście z programu (przywraca normalne litery). Nie należy używać BREAK.

Po każdym naciśnięciu klawiszy 1, 2, 3, 4, w okienku na górze ekranu pokazuje się aktualny zestaw znaków (na który wskazują komórki 23606 i 23607). Niżej podana jest informacja, czy automatyczne przeszukiwanie jest włączone (jeśli tak to A=1) i o tym jak działają klawisze 3 i 4 (jeśli P=1 to zmniejszają lub zwiększają o 8, jeśli P=0 to o 1).

Gdy w okienku zaczną się pokazywać jakieś sensowne kształty, wówczas należy odpowiednimi klawiszami dopasować znak odpowiadający literze A do migającego kwadracika w okienku (najpierw klawiszami 3 i 4 przy P=1, potem przy P=0). Potem należy zanotować format instrukcji SAVE, podany najniżej na ekranie, a następnie wyjść z programu (klawisz 0), aby zapisać zestaw znaków na taśmie wykorzystując wcześniej zanotowaną instrukcję SAVE.

Ilekroć będziemy chcieli użyć wybranego kroju czcionki w swoim programie będziemy musieli wczytać ją instrukcją:
LOAD " " CODE x, gdzie x to adres pamięci, pod którym chcemy umieścić generator znaków. Aby zamienić standardowy zestaw znaków na nowy należy jeszcze wpisać instrukcje:

POKE 23607,INT(x/256)–1 : POKE 23606, 256 ∗INT(x/256)

Teraz wszystkie napisy, jakie będą się pokazywać na ekranie (nawet listing programu), będą napisane nową czcionką.

Michał Szuniewicz

```
   2 CLS
   4 PLOT 0,170: DRAW 255,0
   5 PLOT 0,140: DRAW 255,0
  10 LET SB=23607: LET MB=23606
  20 REM ***********
  30 LET H=91: LET L=0
  35 LET P=1: LET A=0
  40 GO SUB 1000
  50 LET A$=INKEY$
  60 IF A$="" AND A=0 THEN  GO T
O 50
  65 IF A$="9" THEN  LET A=NOT A
  67 IF A$="8" THEN  LET P=NOT P
  70 IF A$="1" THEN  LET H=H-1
  80 IF A$="2" OR A=1 THEN  LET
H=H+1
  81 IF A=1 THEN  LET L=0
  82 IF P=0 THEN  LET S=1
  84 IF P=1 THEN  LET S=8
  90 IF A$="3" THEN  LET L=L-S
 100 IF A$="4" THEN  LET L=L+S
 105 IF A$="0" THEN  POKE SB,60:
 POKE MB,0: STOP
 106 IF L<0 THEN  LET L=255: LET
 H=H-1
 107 IF L>255 THEN  LET L=0: LET
 H=H+1
 108 IF H>252 THEN  LET H=252
 109 IF H<0 THEN  LET H=0
 110 GO SUB 1000
 120 GO TO 50
1000 POKE SB,H
1010 POKE MB,L
1015 PRINT AT 1,0;
1020 FOR T=32 TO 127
1030 PRINT CHR$ T;
1040 NEXT T
1050 POKE 22593,135
1060 POKE MB,0: POKE SB,60
1065 PRINT AT 9,0;"P=";P;"  ";"A=
";A
1069 PRINT AT 10,0;"                "
1070 PRINT AT 10,0;SB;"=";H
1075 PRINT AT 11,0;"                "
1080 PRINT AT 11,0;MB;"=";L
1085 PRINT AT 12,10;"
      "
1090 PRINT AT 12,0;"SAVE "+CHR$
34;"nazwa"+CHR$ 34;"CODE ";(H+1)
*256+L;",768"
1100 RETURN
```

JAK MALOWAĆ? (cz. 1)

Tworząc grafikę na ekranie czy to w Basicu, czy w kodzie maszynowym niejednokrotnie napotykamy na przeszkody typu: jak zapełnić pewien obszar zamknięty?

Zacznijmy od najprostszej figury geometrycznej posiadającej pole — koła. Może tak:

```
10 LET x=128: LET y=88: LET r=30: REM parametry
20 FOR f= 1 TO r
30 CIRCLE x, y, f
40 NEXT f
```

Nie jest to porządnie — pozostają kropki i smugi. Przypomnijmy sobie wiadomości o funkcjach trygonometrycznych i napiszmy (linia 10 bez zmian):

```
20 FOR f = 0 TO 2∗PI STEP PI/180
30 PLOT x, y
40 DRAW r ∗ COS f, r ∗ SIN f
50 NEXT f
```

Też niedobrze. A może skorzystać z równania koła $x^2 + y^2 = r^2$?

```
20 FOR f= –r TO r
30 PLOT x+f, y + SQR ABS (r∗r–f∗f)
40 DRAW0,–2∗SQR ABS (r∗r–f∗f)
50 NEXT f
```

Nareszcie!

Przyczyną tak długich niepowodzeń była mała rozdzielczość Spectrum. Aby uzyskać dobry obraz musieliśmy uciec się do rysowania tylko pionowych (poziomych) linii.

Teraz narysowanie zapełnionej elipsy będzie fraszką.

Znamy równanie:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

więc:

```
10 LET x=128: LET y=88: LET a=40: LET b=30
20 FOR f = –a TO a
30 LET dy = b∗SQR ABS (1–f∗f/(a∗a))
40 PLOT x+f, y+dy
50 DRAW 0, –2∗dy
60 NEXT f
```

a i b to długości półosi elipsy. W podobny sposób zamalować można pozostałe krzywe stożkowe — hiperbolę

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$$

i parabolę

$$y^2 = 2px$$

Teraz kwadrat. Zadanie raczej dla przedszkolaka:

```
20 FOR f = –r TO r
30 PLOT x+ f, y–r
40 DRAW 0,2∗r
50 NEXT f
```

W przypadku prostokąta wystarczy poróżnić r w liniach 20 i 30 (poziom) z r w linii 40 (pion).

W przypadku rombu:

```
10 LET x=128: LET y=88, LET r=30: LET a=PI/3
20 FOR f = –r TOr
30 PLOT x+f, y– r
40 DRAW r∗COS a, r∗SIN a
50 NEXT f
```

Wprowadzamy tu kąt nachylenia boku (a).

A jak zrobić równoległobok? — Tak jak z kwadratu prostokąt.

Tu kończą się możliwości Spectrum w dokładnym zapełnianiu figur. Nie da się szybko i dokładnie zapełnić ani w/w figur obróconych o kąt ostry, ani innych. Rozwiązanie problemu — za miesiąc.

Marcin Przasnyski

9888

Liczba ta jest nazwą, a jednocześnie oznaczeniem początku pewnego użytecznego programu narzędziowego dla ZX Spectrum.

Jedną z wad ZX Spectrum jest niezbyt wygodna edycja tekstu programu. Napisany przeze mnie program stanowi próbę ułatwienia tej operacji.

Program należy wpisać, przetestować i nagrać na kasetę. W razie potrzeby dołączamy „9888" do pisanego programu przy pomocy instrukcji MERGE. Trzeba jednak pamiętać, że program główny nie może zawierać linii o numerach wyższych od 9888.

Po wykonaniu

GO TO 9888

ukazuje się krótkie menu funkcji programu — jest on gotowy do użytku. Oto jego możliwości:

1 — POLICZ P,O

Komputer oblicza liczbę linii zawartych pomiędzy numerami P i O włącznie.

2 — PRZENUMERUJ P, O, N, K

Przenumerowanie fragmentu programu zawartego między liniami P i O. Pierwsza linia fragmentu otrzymuje nowy numer N, zaś numeracja odbywa się z krokiem K. Nie są zmieniane parametry odwołań GO TO i GO SUB.

3 — PRZESUŃ P, O, N

Fragment programu pomiędzy liniami P i O otrzymuje numery zwiększone o N (N może być ujemne).

4 — ZNAJDŹ X, P, O

Wyszukanie wszystkich fragmentów tekstu X w liniach programu pomiędzy P i O.

5 — ZAMIEŃ X = Y, P, O

Zamiana wszystkich występujących pomiędzy liniami P i O fragmentów tekstu X na tekst Y. Zarówno X, jak i Y mogą mieć do 10 znaków długości — pozostałe znaki są ignorowane.

6 — STOP

Powrót do BASIC-a.

Grzegorz Waligórski

```
9888 REM EDYTOR 9888
9890 PRINT #0;"1-POLICZ         3-P
RZESUN 5-ZAMIEN2-PRZENUMERUJ 4-Z
NAJDZ  6-STOP"
9892 LET x$=INKEY$
9893 IF x$="1" THEN GO TO 9900
9894 IF x$="2" THEN GO TO 9910
9895 IF x$="3" THEN GO TO 9920
9896 IF x$="4" THEN GO TO 9930
9897 IF x$="5" THEN GO TO 9960
9898 IF x$="6" THEN STOP
9899 GO TO 9892
9900 REM policz linie od linii p
     do linii o
9901 INPUT "POLICZ ";p;",";o
9902 IF p<=0 OR o<=0 OR p>o THEN
 GO TO 9901
9904 LET n=0: GO SUB 9990
9906 IF PEEK (s+1)+256*PEEK s>o
THEN CLS : PRINT "LICZBA LINII:
";n: GO TO 9888
9908 LET n=n+1: LET s=s+PEEK (s+
2)+256*PEEK (s+3)+4: GO TO 9906
9910 REM przenumeruj linie od p
     do o od numeru n z
     krokiem k
9912 INPUT "PRZENUMERUJ ";p;",";
o;",";n;",";k
9914 IF p<=0 OR o<=0 OR n<=0 OR
k<=0 OR p>o THEN GO TO 9912
9916 LET n0=n: GO SUB 9990
9918 IF PEEK (s+1)+256*PEEK s>o
THEN LIST n0: GO TO 9888
9919 LET b=INT (n/256): POKE s,b
: POKE s+1,n-256*b: LET n=n+k: L
ET s=s+PEEK (s+2)+256*PEEK (s+3)
+4: GO TO 9918
9920 REM przesun linie od p do o
     o liczbe n
9921 INPUT "PRZESUN ";p;",";o;",
";n
9922 IF p<=0 OR o<=0 OR p>o OR p
+n<=0 OR o+n>=9900 THEN GO TO 99
21
9924 GO SUB 9990
9926 LET b=PEEK (s+1)+256*PEEK s
: IF b>o THEN LIST p+n: GO TO 98
88
9928 LET b=b+n: LET n0=INT (b/25
6): POKE s,n0: POKE s+1,b-256*n0
: LET s=s+PEEK (s+2)+256*PEEK (s
+3)+4: GO TO 9926
9930 REM znajdz ciag znakow x$
     w liniach od p do o
9932 DIM v(10)
9934 INPUT "ZNAJDZ "; LINE x$;",
";p;",";o
9936 LET n0=LEN x$: IF n0=0 OR n
0>10 OR p<=0 OR o<=0 OR p>o THEN
 GO TO 9934
9938 FOR i=1 TO n0: LET v(i)=COD
E x$(i): NEXT i: CLS : PRINT "ZN
ALAZLEM ";x$;" W LINIACH:": GO S
UB 9990
9940 LET s=s+3: LET b=PEEK (s-1)
+256*PEEK s-1
9942 FOR i=1 TO b: IF PEEK (s+i)
<>v(1) THEN GO TO 9952
9944 LET i=i-1: IF n0=1 THEN GO
TO 9950
9946 FOR j=2 TO n0: IF PEEK (s+i
+j)<>v(j) THEN LET i=i+1: GO TO
9952
9948 NEXT j
9950 PRINT 256*PEEK (s-3)+PEEK (
s-2): LET i=i+n0
9952 NEXT i
9954 LET s=s+b+2: IF PEEK (s+1)+
256*PEEK s<=o THEN GO TO 9940
9956 GO TO 9888
9960>REM zamien ciag znakow x$
     na y$ w liniach od p
     do o
9961 DIM v(2,10)
9962 INPUT "ZAMIEN "; LINE x$;"=
"; LINE y$;",";p;",";o
9964 LET n0=LEN x$: IF n0=0 OR n
0>10 OR LEN y$<>n0 OR p<=0 OR o<
=0 OR p>o THEN GO TO 9962
9966 FOR i=1 TO n0: LET v(1,i)=C
ODE x$(i): LET v(2,i)=CODE y$(i)
: NEXT i: GO SUB 9990
9968 LET s=s+3: LET b=PEEK (s-1)
+256*PEEK s-1
9970 FOR i=1 TO b: IF PEEK (s+i)
<>v(1,1) THEN GO TO 9980
9972 LET i=i-1: IF n0=1 THEN GO
TO 9978
9974 FOR j=2 TO n0: IF PEEK (s+i
+j)<>v(1,j) THEN LET i=i+1: GO T
O 9980
9976 NEXT j
9978 FOR j=1 TO n0: POKE s+i+j,v
(2,j): NEXT j: LET i=i+n0
9980 NEXT i
9982 LET s=s+b+2: IF PEEK (s+1)+
256*PEEK s<=o THEN GO TO 9968
9984 LIST p: GO TO 9888
9990 REM szukaj linii p
9992 LET s=PEEK 23635+256*PEEK 2
3636
9994 IF PEEK (s+1)+256*PEEK s>=p
 THEN RETURN
9996 LET s=s+PEEK (s+2)+256*PEEK
 (s+3)+4: GO TO 9994
```

DO GÓRY NOGAMI!

Jeżeli jesteś ciekaw, jak ekran wygląda „do góry nogami", jest na to sposób, o wiele mniej pracochłonny, niż odwracanie telewizora, czy stawanie na głowie.

Wystarczy wpisać program:

```
9000>REM Odwracanie ekranu
9010 REM * by Tomasz Siwak
9020 DATA 17,0,64,33,255,87,26,6
,8,23,203,30.16,251,23,18,19,43,
62,76,186,32,239,1,0,88,33,255,9
0,17,129,1,10,50,232,253,126,2,5
8,232,253,119,3,43,27,122,179,32
,239,201
9030 LET s=0
9040 FOR f=30000 TO 30049
9050 READ d: POKE f,d: LET s=s+d
9060 NEXT f
9070 IF s<>4512 THEN STOP
```

Następnie uruchomić go (instrukcją RUN), a po otrzymaniu komunikatu „OK" (w przypadku otrzymania komunikatu „STOP" należy skorygować dane w linii 9020) wczytać dowolny ekran. Aby ujrzeć ten sam ekran odwrócony, należy wpisać: RANDOMIZE USR 30000 (Warto też PAUSE 0, ponieważ odwrócony ekran zajmuje dwie linie robocze u dołu ekranu, których nie zobaczylibyśmy po otrzymaniu komunikatu). Jeżeli ekran przed odwróceniem zajmuje linie robocze, aby nie uciąć ich można wpisać następujące sekwencje rozkazów:

```
LOAD "nazwa bloku"SCREEN$:
RANDOMIZE USR 30000:
PAUSE 0 .
```

Tomasz Siwak

MICROSOFT BASIC

Ostatnim omawianym interpreterem BASIC-a jest napisany najwcześniej, bo już w 1981 roku, Microsoft BASIC. Jest on jednocześnie najmniej popularnym interpreterem BASIC-a dla komputerów Atari.

Microsoft BASIC jest kompatybilny w jedną stronę z Atari BASIC. Dotyczy to jednak tylko programów napisanych w Atari BASIC-u i zapisanych na dyskietce lub kasecie instrukcją LIST. Programy zapisane przez SAVE lub CSAVE nie są rozumiane przez Microsoft BASIC z powodu użycia innych tokenów. Poważną wadą jest pozostawienie dla użytkownika tylko 21022 bajtów pamięci (przy pracy ze stacją dysków). Interpreter Microsoft jest nieco szybszy (około 1,5 raza), a znacznie przewyższa Atari BASIC w obliczeniach numerycznych (około 5 razy szybszy). Zestaw dostępnych instrukcji jest znacznie większy, lecz pisanie programów utrudnia konieczność wpisywania słów kluczowych bez skrótów i z odstępami.

Uwzględniając powyższe uwagi należy stwierdzić, że Microsoft BASIC powinien być używany jedynie do programów obliczeniowych, w których można efektywnie wykorzystać jego możliwości (przede wszystkim tablice wielowymiarowe i liczby podwójnej precyzji).

INSTRUKCJE REDAKCYJNE

Uruchamianie programów jest bardzo łatwe dzięki dużej liczbie instrukcji pomocniczych. Błędy są sygnalizowane meldunkami określającymi ich rodzaj, a nie tylko kod. Niestety, poprawność linii programu nie jest sprawdzana po jej wprowadzeniu, lecz dopiero podczas wykonywania.

AUTO[m,n]
Automatyczne numerowanie linii programu od linii o numerze m z przyrostem n. Niepodanie wartości jest równoważne instrukcji AUTO 100,10. Jeżeli w pamięci jest już jakiś program, to numeracja rozpoczyna się od linii o numerze większym o n od najwyższego istniejącego numeru linii. Naciśnięcie RETURN po wyświetleniu numeru linii powoduje opuszczenie trybu automatycznej numeracji.

RENUM[k[,m[,n]]]
Przenumerowanie linii programu: k — numer pierwszej „nowej" linii, m — numer pierwszej „starej" linii, n — krok numeracji. Wartościami standardowymi są: k = 10, m = 0, n = 10.

DEL[m][-[n]]
Usunięcie linii programu od linii m lub od linii 0 do linii n lub do końca.

ERROR n
Powoduje wystąpienie w programie błędu o kodzie n. Instrukcja ta jest bardzo użyteczna przy testowaniu programu (pamiętaj o jej późniejszym usunięciu).

TRON
Włącza śledzenie programu: przed wykonaniem każdej linii programu wyświetlany jest jej numer.

TROFF
Wyłącza śledzenie programu.

DEKLARACJE

DEFINT zmienna [,zmienna ...]
Deklaracja zmiennej całkowitej. Zamiast deklaracji można na końcu nazwy zmiennej umieścić znak % (np. A% zamiast DEFINT A, NUM). Liczby całkowite mają zakres od -32768 do +32767.

DEFSNG zmienna [,zmienna ...]
Deklaracja zmiennej pojedyńczej precyzji. Można ją pominąć, gdyż każda zmienna nie mająca na końcu nazwy jednego ze znaków %, # lub $ jest traktowana jako zmienna pojedyńczej precyzji (np. A, NUM zamiast DEFSNG A, NUM). Liczby pojedyńczej precyzji mają zakres od −1.71041E+38 do −2.938737E−39 i od +2.938737E−39 do +1.71041E+38

DEFDBL zmienna [,zmienna ...]
Deklaracja zmiennej podwójnej precyzji. Zamiast deklaracji można na końcu nazwy zmiennej umieścić znak # (np. A#, NUM# zamiast DEFDBL A, NUM). Liczby podwójnej precyzji mają taki sam zakres jak liczby pojedyńczej precyzji, lecz większą ilość cyfr znaczących.

DEFSTR zmienna [,zmienna ...]
Deklaracja zmiennej tekstowej. Zamiast deklaracji można na końcu nazwy zmiennej umieścić znak $ (np. A$, NUM$ zamiast DEFSTR A, NUM).

DEFFN funkcja (zmienna[, zmienna])=wyr.arytm.
Deklaracja funkcji użytkownika. Np. DEFFN A(X,Y) = SQR (X*X+Y*Y).

DIM zmienna (m[,n...])
Deklaracja tablicy liczbowej lub tekstowej. Maksymalny wymiar oraz liczba elementów tablicy (także tekstowej) zależy od wielkości dostępnej pamięci (np. DIM A$ (2,3,10,5),B(3,3,3,2,2) itd.). Proste (jednowymiarowe) zmienne tekstowe nie wymagają deklarowania instrukcją DIM (wystarczy użycie nazwy np. T$ lub deklaracja DEFSTR T).

COMMON ALL lub **COMMON** zmienna [,zmienna...]
Instrukcja zachowująca wartości wszystkich (ALL) lub wymienionych zmiennych w różnych programach, tzn. gdy jeden program jest wywoływany przez drugi instrukcją RUN „D: nazwa" lub RUN „C:".

INSTRUKCJE PROGRAMOWE

RUN n
Uruchomienie programu od linii o numerze n. Oprócz tego możliwe są wszystkie warianty użycia RUN stosowane w Atari BASIC.

IF war.log **THEN** instr.1 [**ELSE** instr.2]
Rozbudowana instrukcja IF/THEN. Gdy warunek logiczny (war.log.) jest spełniony, wykonywana jest instrukcja instr.1, w przeciwnym razie instr.2. Jeżeli ELSE instr.2 zostało opuszczone, to program przechodzi do następnej linii.

ON ERROR n
Skok do linii o numerze n w przypadku wystąpienia błędu podczas wykonywania programu.

RESUME [n/ /**NEXT]**
Umieszczona na końcu procedury obsługi błędu (wywoływanej przez ON ERROR) instrukcja RESUME przekazuje sterowanie programu do linii o numerze n (RESUME n), do linii, w której pojawił się błąd (RESUME) lub do linii następującej po linii, w której pojawił się błąd (RESUME NEXT).

RANDOMIZE n
Użycie tej instrukcji przed RND powoduje generowanie za każdym razem takiego samego ciągu liczb losowych (n jest wartością początkową generatora liczb losowych).

MOVE (m,n,k)
Przemieszczenie bloku danych o długości k bajtów z obszaru pamięci rozpoczynającego się od adresu m do obszaru rozpoczynającego się od adresu n.

CLEAR
Instrukcja zerowania zmiennych numerycznych i tekstowych — odpowiada instrukcji CLR w Atari BASIC.

AFTER n **GOTO** m
Powoduje zatrzymanie wykonywania programu na okres n/50 sekund, a następnie wykonanie skoku do linii o numerze m.

OPTION BASE 0 lub **OPTION BASE 1**
Określa najmniejszy indeks tablicy. Np. sekwencja OPTION BASE 1:DIM A(3) powoduje zadeklarowanie tablicy trzyelementowej (od A(1) do A (3)), a sekwencja OPTION BASE 0:DIM A(3) powoduje zadeklarowanie tablicy czteroelementowej (od A(0) do A(3)). Może być użyta w programie tylko jeden raz.

OPTION CHR1 lub **OPTION CHR2**
Rezerwuje obszar pamięci o wielkości 1024 (CHR1) lub 2048 (CHR2) bajtów dla zestawu znaków zaprojektowanego przez użytkownika. Kasuje się instrukcją **OPTION CHR0.**

OPTION PLM1 lub **OPTION PLM2**
Rezerwuje obszar pamięci dla grafiki graczy i pocisków (P/MG). Przy rozdzielczości jednowierszowej należy użyć PLM1 (2048 bajtów), a przy dwuwierszowej — PLM2 (1024 bajty). Kasowanie instrukcją **OPTION PLM0.**

OPTION RESERVE n
Rezerwuje obszar bajtów do wykorzystania przez użytkownika, np. na procedury w języku maszynowym. Każda następna instrukcja OPTION RESERVE kasuje działanie poprzedniej.

INSTRUKCJE GRAFICZNE

CLS [n]
Czyszczenie ekranu i wpisanie do rejestru koloru tła wartości n. W Atari BASIC odpowiada to instukcjom? CHR$(125):POKE 712,n.

AT (x,y)
Występuje w instrukcjach PRINT, INPUT i GET i określa miejsce zapisu lub odczytu. Liczby w nawiasie określają współrzędne punktu na ekranie lub numer sektora i bajtu na dyskietce.

TAB (n)
Ustawia kursor w kolumnie o numerze n w aktualnie pisanej linii lub w następnej, jeżeli wartość n jest mniejsza od bieżącej pozycji kursora. Użycie dozwolone tylko w instrukcji PRINT.

SPC n
Użyta z instrukcją PRINT powoduje wyświetlenie n spacji a użyta z PRINT # — zapisanie n spacji na urządzenie zewnętrzne.

PLOT x1,y1 [**TO** x2,y2] [**TO** ...]
Rysuje na ekranie punkt o współrzędnych x1,y1 i odcinek do punktu o współrzędnych x2,y2 oraz ewentualnie dalsze odcinki. Instrukcji PLOT x1,y1

TO x2,y2 odpowiada w Atari BASIC sekwencja PLOT x1,y1: DRAWITO x2,y2.

FILL x1,y1 **TO** x2,y2
Wypełnia kolorem obszar na ekranie ograniczonym z góry i z dołu punktami o podanych współrzędnych. Z boku wypełniany obszar musi być ograniczony narysowanymi wcześniej odcinkami.

PRINT USING „format"; zmienna [,zmienna...]
Określenie formatu wyświetlanej informacji. Ciąg „format" zawiera znaki oznaczające sposób wyświetlenia:
— znak zmiennej (cyfra lub litera)
. — kropka dziesiętna
, — przecinek
+ —przed liczbą dodatnią +, przed ujemną –
– — przed liczbą ujemną –, liczba dodatnia bez znaku
$ — znak dolara jako pierwszy
Np. PRINT USING „##,###.##", 1234,567.8 da w efekcie: 1,234.00 567.80

INSTRUKCJE WEJŚCIA/WYJŚCIA

MERGE „C:" lub **MERGE** „D: nazwa — pliku"
Wczytuje program z podanego urządzenia zewnętrznego i dołącza go do znajdującego się już w pamięci. Wczytywany program musi być zapisany instrukcją LIST (jak przy ENTER w Atari BASIC).

VERIFY „C:" lub **VERIFY** „D: nazwa — pliku"
Porównuje program zapisany na kasecie lub dyskietce z programem zawartym w pamięci (weryfikuje zapis).

NAME „D: stara — nazwa" **TO** „nowa — nazwa"
Zmienia nazwę pliku zapisanego na dyskietce.

KILL „D:nazwa — pliku"
Usuwa z dyskietki plik (program lub dane) o podanej nazwie.

LOCK „D:nazwa — pliku"
Zabezpiecza plik dyskowy o podanej nazwie. Zabezpieczenie chroni przed skasowaniem, zmianą nazwy i zapisaniem innego pliku o tej samej nazwie, nie chroni przed skasowaniem pliku podczas formatowania dyskietki.

UNLOCK „D:nazwa — pliku"
Odbezpiecza plik dyskowy o podanej nazwie.

OPEN #Fn, „urz: [nazwa — pliku]" operacja
Otwarcie kanału o numerze n do odsługi urządzenia zewnętrznego o kodzie urz. Dla stacji dysków konieczne jest także podanie nazwy pliku. Kody urządzeń są takie same jak w Atari BASIC. Rodzaj przeprowadzanej operacji musi być podany przy pomocy jednego z czterech określeń:
— **INPUT** — operacja wejścia (odczyt),
— **OUTPUT** — operacja wyjścia (zapis),
— **UPDATE** — operacja wejścia/wyjścia (wymiana danych),
— **APPEND** — operacja dołączenia (dopisanie na końcu pliku).

INPUT [# n] [„tekst"] [,**AT** (x,y);] zmienna [,zmienna...]
Wprowadzenie rekordu (ciągu danych zakończonego znakiem RETURN) z kanału o numerze n, a gdy n nie zostało podane, to z klawiatury. Jeżeli został podany „tekst", to jest on wyświetlany na ekranie, w przeciwnym wypadku wyświetlany jest znak „?". Przy wprowadzaniu danych z dyskietki można podać konkretny sektor i bajt (zob.. opis instrukcji AT).

FUNKCJE

RND(n)
Podaje liczbę losową. Dla n=0 jest to liczba rzeczywista z przedziału <0,1), a dla n>0 — liczba całkowita z przedziału (0,n>. Podana liczba n musi być zerem lub liczbą naturalną.

VARPTR (nazwa — zmiennej)
Podaje adres początkowy zmiennej w pamięci komputera (jak ADR w Atari BASIC).

POS(0)
Podaje numer kolumny ekranu, w której znajduje się kursor.

SCRM(x,y)
Podaje kod znaku (w trybach tekstowych) lub numer koloru (w trybach bitowych) znajdującego się w punkcie ekranu o podanych współrzędnych.

EOF(n)
Przybiera wartość –1, jeżeli został osiągnięty koniec pliku w kanale o numerze n oraz 0 w przeciwnym przypadku.

ERR
Podaje kod błędu, który wystąpił (1 — bez błędu).

ERL
Podaje numer linii, w której wystąpił błąd.

TIME
Podaje zawartość komórek zegara systemowego — odpowiada to wrażeniu PEEK (20)+256*PEEK(19)+65 536*PEEK(18).

TIME$
Podaje czas w/g zegara systemowego w formacie „hh:mm:ss"

INKEY$
Podaje znak ASCII odpowiadający ostatnio naciśniętemu klawiszowi.

**LEFT$**(t$,n)
Podaje część ciągu t$ o długości n znaków rozpoczynająca się od pierwszego znaku.

**RIGHT$**(t$,n)
Podaje część ciągu t$ o długości n znaków kończącą się na ostatnim znaku

**MID$**(t$,m,n)
Podaje część ciągu t$ o długości n znaków rozpoczynającą się od m-tego znaku.

INSTR([n,]t1$,t2$)
Podaje pozycję pierwszego znaku zmiennej tekstowej t2$ w zmiennej tekstowej t$1 lub 0, gdy t2$ nie jest częścią t1$. Przeszukiwanie ciągu t1$ rozpoczyna się od n-tego znaku.

STRINGS (n,t$/m)
Tworzy ciąg złożony z n powtórzeń ciągu t$ lub znaku o kodzie ASCII m.

m **AND** n
Podaje bitowy iloczyn logiczny stałych lub zmiennych liczbowych m i n. W wyrażeniach logicznych działa jak w Atari BASIC.

m **OR** n
Podaje bitową sumę logiczną stałych lub zmiennych liczbowych m i n. W wyrażeniach logicznych działa jak w Atari BASIC.

NOT n
Podaje bitową negację logiczną stałej lub zmiennej liczbowej n. W wyrażeniach logicznych działa jak w Atari BASIC.

m **XOR** n
Podaje bitową alternatywę logiczną (EXCLUSIVE-OR) stałych lub zmiennych liczbowych m i n.

OBJAŚNIENIA

W standardzie Microsoft BASIC nie występują (zostały usunięte lub zastąpione innymi) następujące instrukcje: **BYE, CLR, COM, DEG, DRAWTO, ENTER, LOCATE, LPRINT, POP, POSITION, RAD, TRAP, XIO** oraz następujące funkcje: **ADR, CLOG, PADDLE, PTRIG, STRIG.**

INSTRUKCJE I FUNKCJE USUNIĘTE

W opisie instrukcji i funkcji zastosowane zostały następujące symbole:
[..] — użycie możliwe, ale niekonieczne (zależnie od potrzeby);
../.. — należy użyć jeden z dwóch wariantów;
... — dozwolona dowolna ilość powtórzeń elementów;
k,m,n,x,y — stałe lub zmienne liczbowe;
t$,t1$,t2$ — stałe lub zmienne tekstowe;
„tekst" — stała tekstowa;
zmienna — zmienna liczbowa lub tekstowa.

Wojciech Zientara

DUSZEK RAZ JESZCZE

Przeczytałem niedawno w „Bajtku artykuł o wykorzystaniu Player/Missile Graphics na Atari. Jednak żaden z proponowanych w artykule sposobów uzyskania ruchu pionowego nie zadowolił mnie, dlatego napisałem program symulujący rejestr pionowy dla gracza 0.

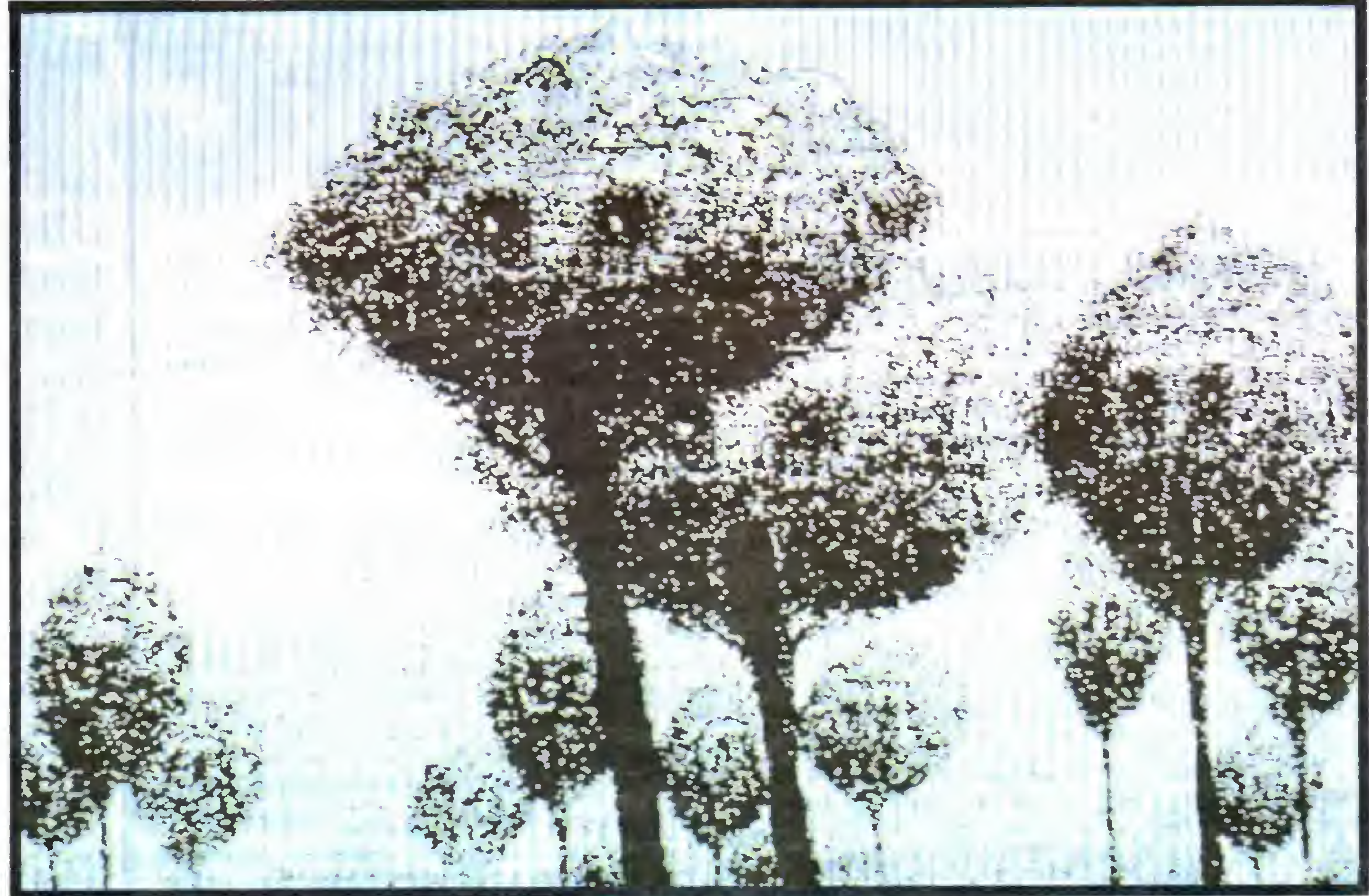
Program nie jest zbyt doskonały, główną jego wadą jest to, że obsługuje jedynie gracza 0. Należy go więc traktować raczej jako zasygnalizowanie pomysłu niż jako gotowy produkt. Program oparty jest o przerwanie VBLK, aby go wykorzystać należy:
1. Dołączyć go jako podprogram do własnego programu.
2. Postać gracza 0 zapisać od adresu 1600, maksymalnie może on mieć 190 bajtów.
3. Do komórki 1538 wpisać instrukcją POKE długość gracza (liczbę bajtów wzoru).
4. Po zmienną PMB podstawić adres bazowy podzielony przez 256 (czyli wartość zapisywaną do rejestru PMBASE).
5. Uruchomić podprogram.

Od tej chwili komórka o adresie 1537 pełni rolę rejestru pionowego.

Leszek Paprocki

```
30000 FOR N=1559 TO 1596:READ Q:POKE N
,Q:NEXT N
30001 DATA 174,0,6,172,2,6,169,0,157,4
3,2,202,136,208,249,173,1,6,141,0,6,17
0,172,2,6,185
30002 DATA 64,6,157,43,2,202,136,208,2
46,76,98,228
30004 POKE 1569,PMB+4:POKE 1589,PMB+4
30005 POKE 1568,0:POKE 1588,0
30006 FOR N=1536 TO 1545:READ Q:POKE N
,Q:NEXT N
30007 DATA 104,160,23,162,6,169,7,76,9
2,228
30008 Q=USR(1536)
30009 POKE 54286,64
```

KATALOG



Grafika: A. Pągowski

Z pewnością często mieliście ochotę zrobić porządek na swoich kasetach z programami i równie często rezygnowaliście z tego mając w perspektywie pracowite spisywanie z ekranu danych uzyskanych przez komendę CAT. Jak dobrze byłoby zwalić tę robotę na komputer. Mógłby on wtedy sam przejrzeć kasetę, sprawdzić, czy jakieś zbiory nie powtarzają się, wydrukować spis na drukarce. Niestety, CAT wysyła wyniki tylko na ekran i odzyskanie ich stamtąd wymaga wiele wysiłku. Komendy tej nie można również przerwać programowo a jedynie klawiszem ESC.

Równie często ambitniejsi użytkownicy Amstradów i Schneiderów usiłują uzyskać dane o długości programu, miejscu jego ładowania, czy adresie autostartu. Dane te są niezbędne na przykład do przenoszenia programów maszynowych z kasety na dyskietkę. I tutaj komenda CAT nie zdaje się na nic.

Zamieszczony obok listing programu w BASIC-u umożliwia dodanie do systemu nowej komendy RSX o nazwie CATALOG, która rozwiązuje nasze problemy.

Po uruchomieniu programu, jeżeli nie pojawi się na ekranie żaden komunikat o błędzie w linii DATA (spowodowanym błędnym przepisaniem) w obszar pamięci od adresu & A400 zostanie załadowany i zainicjowany program maszynowy komendy CATALOG. Program w BASIC-u można teraz skasować (przez NEW).

Wywołanie komendy ma następującą postać:

| CATALOG, @a$, @a1%, @a2%, @a3%, @4%, @a5%, @a6%, @a7% gdzie zmienne a$, a1%, a2%... a7% muszą mieć uprzednio nadane wartości (w przeciwnym przypadku pojawi się komunikat „Improper argument"). Ich nazwy nie muszą być oczywiście takie jak wyżej. Ważne są jednak typy: pierwsza musi być łańcuchowa — pozostałe całkowite. Symbol „@" oznacza, że do komendy przekazywany jest adres zmiennej, a nie jej wartość. Wartości nadane zmiennym całkowitym przed wywołaniem są obojętne. Natomiast a$ powinien być przypisany łańcuch zawierający dowolne znaki, lecz o długości co najmniej 16. Komenda CATALOG uruchamia silnik magnetofonu i wczytuje nagłówek najbliższego bloku (trzeba więc pamiętać o wciśnięciu klawisza PLAY przed wywołaniem). Dane o bloku kopiowane są do parametrów w następujący sposób:

- **a$** — zawiera nazwę zbioru, do którego należy blok; jeżeli wartość a$ przed wywołaniem była łańcuchem krótszym niż długość nazwy, to nadmiarowe znaki zostaną obcięte; jeżeli łańcuch był dłuższy niż nazwa, to pozostałe miejsca zostaną zapełnione spacjami
- **a1%** — zawiera numer bloku
- **a2%** — zawiera typ zbioru: 0 — BASIC, 1 — BASIC chroniony (SAVE „nazwa", P), 2 — program w języku maszynowym, 22 — zbiór ASCII
- **a3%** — adres ładowania bloku; adres ładowania pierwszego bloku w zbiorze jest zarazem adresem ładowania całego zbioru (dla ASCII = 0, dla Basica = 368)
- **a4%** — długość bloku; dla wszystkich bloków z wyjątkiem ostatniego prawie zawsze 2048 = 2 KB
- **a5%** — długość całego zbioru, do którego należy blok
- **a6%** — adres autostartu; od tego adresu rozpocznie się wykonywanie programu maszynowego załadowanego komendą RUN „nazwa"
- **a7%** — znaczniki bloku: a7% zawiera wartość całkowitą dwubajtową: starszy bajt różny od zera oznacza pierwszy blok zbioru, młodszy bajt różny od zera — ostatni blok; wartość tę najlepiej obserwować jako HEX$ (a7%,4) dwie pierwsze cyfry to starszy bajt, dwie pozostałe — młodszy (cyfry hex)

Jeżeli komenda CATALOG zostanie wywołana bez parametrów, jej wykonanie zakończy się natychmiast, bez wczytywania nagłówka. Jeżeli podanych będzie mniej niż 8 parametrów, to komenda zadziała normalnie z tym, że informacje dla brakujących zmiennych zostaną zignorowane. Niezgodność typów parametrów (pierwszy zawsze łańcuch, pozostałe całkowite) zasygnalizowana będzie komunikatem „Parametr error" i wykonanie zostanie przerwane.

Sergiusz Wolicki

```
10 DATA 210ea40109a4c3d1bc12a4c31aa408
20 DATA fca609a4434154414c4fc700b7c849
30 DATA 4fc50d2812dd6e00dd66012b7e3dd0
40 DATA c2c1a4dd23dd2318ebdd6e00dd66b8
50 DATA 012b7efe02c2c1a421e2a4114000c9
60 DATA 3e2cdde5cda1bcdde1d2cea4dd6ea3
70 DATA 00dd66017eb7281f47235e23562122
80 DATA e2a40e107eb720023e2012231305a6
90 DATA 28090d20f13e20121310fcc1af2a78
100 DATA f2a467cdafa42af4a467cdafa42a90
110 DATA f7a4cdafa42af5a4cdafa42afaa466
120 DATA cdafa42afca4cdafa43af3a46f3a84
130 DATA f9a467cdafa4c90d281cdd2bdd2b4e
140 DATA ebdd6e00dd6601732372c921d0a4e0
150 DATA 7eb72806cd5abb2318f6c1c95061b1
160 DATA 72616d657465722065727276f720d47
170 DATA 0a0000000000000000000000000a
180 '
190 'CATALOG
200 '(c) S.W.
210 '
220 MEMORY &A3FF
230 b=-1:FOR k=0 TO 16:READ a$
240 s=0:FOR i=0 TO 13
250 byte=VAL("&"+MID$(a$,i*2+1,2))
260 POKE &A400+k*14+i,byte:s=s+byte
270 NEXT i
280 IF RIGHT$(HEX$(s,2),2)<>UPPER$(RIGHT$(a$,2)) THEN PRINT "Blad w linii ";(k+1)*10:b=0
290 NEXT k
300 IF b THEN CALL &A400
```

Co piszczy pod klawiaturą?

PROCEDURA KOŃCZĄCA

(część 12 — trzynasta byłaby feralną)

(ciąg dalszy nie nastąpi)

Wizja nadchodzącego, jak zwykle pechowego trzynastego odcinka cyklu sprawiła, że w autorze obudziło się sumienie i postanowił nie męczyć już dłużej czytelników widokiem monotonnych tabelek, wypełniających cenną przestrzeń Klanu Amstrad-Schneider. Jednakże sumienie nie zadowoliło się samym postanowieniem i zażądało pokuty, w ramach której zobligowało autora do odpowiedzi na dwa pytania: „po co to wszystko było drukowane?" i „co z tego ma Czytelnik?". Kończąc cykl, skruszony autor odpowiada...

Dokładna znajomość systemu operacyjnego mikrokomputera pozwala nie tylko na oszacowanie jego możliwości, ale bardziej zaawansowanym użytkownikom umożliwia przystosowanie sprzętu do konkretnych i indywidualnych wymogów m.in. poprzez modyfikację lub rozbudowę procedur systemowych. Jednakże działalność taka jest możliwa tylko wtedy, gdy dysponuje się listą procedur i opisem ich działania... a także potrafi się programować w języku maszynowym.

Zamiarem autora nie było prowadzenie kursu programowania lecz dostarczenie niezbędnej dokumentacji. Wprawdzie niektóre procedury mogą być wywoływane bezpośrednio z poziomu BASIC-a i znacznie upraszczają pisanie programów (np. **CALL & BBBA** jest odpowiednikiem ciągu rozkazów: **PAPER 0:PEN 1:ORIGIN 0, 0: WINDOW # 0, 0, 639, 0, 399** lub **CAll & BB18** zastąpi ciąg **A$=" ":WHILE A$=" ":A$=INKEY$:WEND)** lecz większość procedur może być uaktywniana dopiero po ustaleniu pewnych stanów wejściowych w wymaganych dla danej procedury rejestrach. Taka sytuacja wymusza wprawdzie na programiście znajomość sposobu działania procedur systemu, ale jednocześnie pozwala na wprowadzanie pożądanych zmian.

Zacznijmy od najprostszych przykładów:

Procedura Screen Mode Set (ustalenie trybu pracy ekranu monitora) & BCØE w momencie uruchomienia odczytuje informację z rejestru A (akumulatora), mówiącą o wybranym trybie pracy. Pisząc program w języku maszynowym trzeba przed wywołaniem procedury wpisać do akumulatora Ø,1 lub 2. Dla trybu MODE 2 wygląda to następująco:

kod hexa	mnemonik	
3EØ2	LD A 2	wprowadź 2 do rej.A
CDØEBC	CALL BCØE	wywołaj procedurę Screen Mode Set
C9	RET	powrot do programu

Procedura drukowania znaku na ekranie & BB5A wymaga podania w rejestrze A kodu znaku, ale jednak wcześniej trzeba również określić tryb pracy ekranu i pozycję na której ma być ten znak drukowany, czyli podać parametry wejściowe wymagane w procedurach odpowiednio & BCØE i & BB75 (patrz odcinek 7 i 4 cyklu). Wygląda to tak:

3E02	LD A,2	wprowadź 2 do akumulatora
CD0EBC	CALL BC0EH	wywołaj procedurę SMS
260A	LD H,10	wprowadź numer kolumny (10)
2E10	LD L,16	wprowadź numer wiersza (16)
CD75BB	CALL BB75H	wywołaj procedure & BB75 (pozycja)
3E42	LD A,66	wprowadź kod litery B do rej. A
CD5AB3	CALL BB5AH	wyświetl znak B na ekranie
C9	RET	wróć do programu głównego

Modyfikacje procedur wymagają oczywiście bardziej skomplikowanych zabiegów, jednak ze względu na charakter cyklu i szczupłość miejsca podam jedynie mechanizm przeprowadzania modyfikacji. Załóżmy, że nie wystarcza nam 7-bitowy sprzęg w CPC, ponieważ chcemy wykorzystywać pełne możliwości przyłączanej do niego drukarki. Jednocześnie wiemy, iż drukarka ma możliwość ustawienia 8-go bitu po odebraniu określonego znaku lub sekwencji znaków (w przypadku DMP 2000 jest to ESC ">" i skasowania 8-go bitu po odebraniu innej sekwencji (ESC „="). Mamy również świadomość, że procedura wywoływana przez & BD2B ograniczona jest właśnie 7-bitowym sprzęgiem. Rozwiązanie problemu polega na napisaniu programu, który najpierw sprawdzi, czy wysyłany do drukarki znak ma ustawiony ósmy bit. Jeśli znak ma ósmy bit=0 to powinien być od razu przesłany do drukarki. W przeciwnym przypadku najpierw powinien być przesłany ESC ">", następnie znak i ESC "=". Tak napisany program trzeba ulokować w pamięci RAM poza obszarem dostępnym dla BASIC'a (MEMORY HIMEM-długość programu) dołączając wcześniej do niego skopiowane z adresów & BD2B, & BD2C i & BD2D instrukcje skoku do procedury wydruku znaku w pamięci ROM. Teraz wystarczy wpisać pod adres & BD2B kod C3 (skok pod dwubajtowy adres), pod & BD2C młodszy bajt a pod & BD2D starszy bajt adresu początku naszego programu i po uruchomieniu drukować aż do chr$ (225). Czytelników oczekujących gotowego rozwiązania informuję, że programy takie były już publikowane w naszej prasie (o ile dobrze pamiętam pionierem był pan Wojciech Wojtanowski z programem opublikowanym w Przeglądzie Technicznym).

Namawiając do dalszej samodzielnej pracy podaję dwa krótkie programy umożliwiające odczyt na ekranie monitora zawartości pamięci ROM (dolnej) w postaci kodów ASCI i heksadecymalnym. Niewielkie zmiany programu pozwolą również na odczyt zawartości pamięci ROM górnej. W tym celu należy:

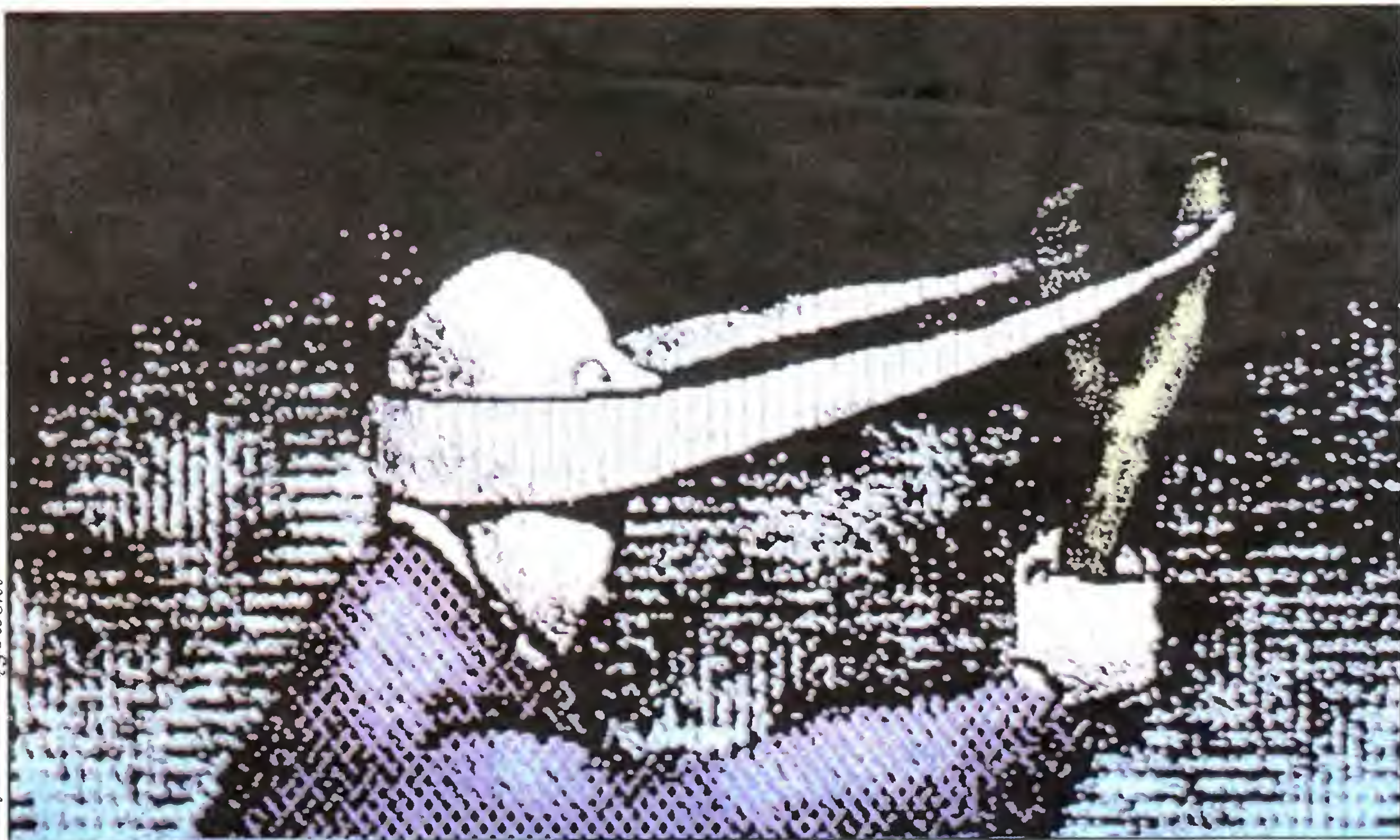
w linii 10 — wpisać ...ROM górny... & C000 do & FFFF
w linii 40 — wpisać ... górnej...
w linii 110 — zamiast cd,06,bd wpisać cd,00,bd i zamiast 00, 00,11 wpisać 00,c0,11
w linii 140 — zamiast (i- 6000) wpisać i+ 6000).

Zachęcam do modyfikacji programów w celu uzyskania możliwości odczytu ograniczonej przestrzeni adresowej (np. zajmowanej tylko przez jedną procedurę) i organizacji wydruku na drukarce. Finis coronat opus!

Wojciech Ziółek

```
10 'dump ACII ROM dolny (od adresu &0000
 do &3FFF)
20 MODE 2:CLS
30 WINDOW #0,1,80,5,25:WINDOW #1,1,80,1,
4
40 PRINT #1,TAB(32) "Zawartosc dolnej pa
mieci ROM":PRINT #1
45 PRINT #1,TAB(4)  CHR$(150);"------>";
50 FOR i=15 TO 78 STEP 4:PRINT #1,TAB(i)
 "+"+HEX$((i-15)/4);:NEXT:PRINT #1,TAB(4
) CHR$(145)
60 MEMORY &6000
70 FOR i=&A000 TO &A010
80 READ a$
90 POKE i,VAL("&H"+a$)
100 NEXT i
110 DATA f3,cd,06,b9,21,00,00,11,00,60,0
1,ff,3f,ed,b0,c9,00
120 CALL &A000
130 FOR i=&6000 TO 40960
140 g$=RIGHT$("000"+HEX$(i-&6000),4)
150 IF INT(i/16)*16=i THEN PRINT g$;"
-->     ";
160 a=PEEK(i)
170 IF (a>31 AND a<127) OR a>159 THEN PR
INT CHR$(a):"    ";ELSE PRINT ".";"    ";
190 NEXT i
```

```
10 'dump hexa ROM dolny (od adresu &0000
 do &3FFF)
20 MODE 2:CLS
30 WINDOW #0,1,80,5,25:WINDOW #1,1,80,1,
4
40 PRINT #1,TAB(32) "Zawartosc dolnej pa
mieci ROM":PRINT #1
45 PRINT #1,TAB(4)  CHR$(150);"------>";
50 FOR i=15 TO 78 STEP 4:PRINT #1,TAB(i)
 "+"+HEX$((i-15)/4);:NEXT:PRINT #1,TAB(4
) CHR$(145)
60 MEMORY &6000
70 FOR i=&A000 TO &A010
80 READ a$
90 POKE i,VAL("&H"+a$)
100 NEXT i
110 DATA f3,cd,06,b9,21,00,00,11,00,60,0
1,ff,3f,ed,b0,c9,00
120 CALL &A000
130 FOR i=&6000 TO 40960
140 g$=RIGHT$("000"+HEX$(i-&6000),4)
150 IF INT(i/16)*16=i THEN PRINT g$;"
-->     ";
160 a=PEEK(i)
170 a$=RIGHT$("00"+HEX$(a),2)
180 PRINT a$;"    ";
190 NEXT i
```



Grafika: A. Pągowski

SUPER EXPANDER PLUS

Za jeden z większych sukcesów finansowych uchodzi niewątpliwie sprzedanie tego samego wyrobu co najmniej dwa razy; sukces jest znacznie większy, gdy dany towar sprzedamy minimum trzykrotnie. Wie o tym najlepiej firma Commodore, gdyż swoje dwa wyroby — BASIC V3.5 i V4.0 sprzedała już znacznie większą liczbę razy.

Pierwszy raz w komputerach PET (Personal Electronic Translator). Drugi i trzeci — w komputerach C-16 i 116. Czwartym razem był to Commodore PLUS/4, piątym C-128 i szóstym — SUPER EXPANDER, program zawarty w module i rozszerzający BASIC Commodore 64 do możliwości mniej więcej BASIC C-16. Omawiany w artykule program otrzymaliśmy do testowania od pana Bolesława Chapińskiego z USA (SUPER EXPANDER) oraz od panów Krzysztofa Gajewskiego i Bogusława Radziszewskiego (SUPER EXPANDER PLUS), twórców WARSAW BASIC.

O samym programie SUPER EXPANDER można powiedzieć sporo dobrego — ułatwia on znacznie programowanie sprite'ów, grafiki wysokiej rozdzielczości, muzyki, ponadto zawiera w sobie funkcje przydatne do odczytywania położenia pióra świetlnego, joysticka oraz wiosełek. Z funkcji typowo edytorskich SUPER EXPANDER zawiera jedynie KEY, za pomocą której można przypisywać klawiszom funkcyjnym f1-f8 poszczególne rozkazy i instrukcje.

GRAFIKA

jest najmocniejszym punktem tego kolejnego rozszerzenia BASIC Commodore 64. SUPER EXPANDER zawiera następujące instrukcje: BOX, CHAR, CIRCLE, COLOR, DRAW, GRAPHIC, GSHAPE, LOCATE, PAINT, SCALE, SCNCLR, SSHAPE oraz (do obsługi sprite'ów) COLINT, MOVSPR, SPRCOL, SPRDEF, SPRITE i SPRSAV. Funkcjami graficznymi są tu RCLR, RDOT oraz (dla sprite'ów) RBUMP, RSPCOL, RSPPOS, i RSPR.

Instrukcja GRAPHIC służy nam do włączania odpowiedniego trybu graficznego wśród których mamy: dwukolorowy tryb o rozdzielczości 320 * 200 punktów, tryb wielokolorowy (160 * 200 punktów i 4 kolory) oraz tryb mieszany (split) gdzie możliwe jest jednoczesne wyświetlanie obrazu graficznego i tekstu. BOX, CIRCLE i DRAW umożliwiają wykreślanie w zasadzie dowolnych figur geometrycznych, oraz linii (DRAW), pojedynczych punktów itp. Tekst może być także przenoszony na ekran graficzny za pomocą CHAR omawianego podobnie jak i RDOT, SSHAPE oraz GSHAPE szerzej w BAJTKU 10/87. COLOR służy do przypisywania wybranych kolorów zarówno dla ekranu tekstowego jak i graficznego. LOCATE — umieszcza kursor graficzny (w postaci pojedynczego punktu graficznego) w żądanym miejscu ekranu. SCNCLR czyści wszystkie ekrany, PAINT pozwala wypełnić daną figurę określonym kolorem, SCALE natomiast umożliwia dobranie odpowiedniej skali dla naszego rysunku. SSHAPE i GSHAPE pozwalają na przypisanie zmiennej tekstowej np. A$ wybranego wycinka ekranu graficznego i jego ponowne odtworzenie. Funkcja RCLR umożliwia odczytanie koloru w dowolnie wybranym miejscu i na dowolnie wybranym ekranie. RGR informuje, który z trybów graficznych jest aktualnie włączony, a RDOT podaje aktualne współrzędne kursora graficznego lub też koloru przypisanego punktowi w którym kursor ten się znajduje.

SUPER EXPANDER ułatwia znacznie także programowanie sprite'ów. SPRDEF włącza specjalny edytor na którym zaprojektowanie wzoru czy wzorów dla naszych sprite'ów jest dziecinnie proste. SPRSAV umożliwia ich zapis na taśmie bądź dysku. SPRITE — włącza i określa parametry danego (danych) sprite'ów, MOVSPR natomiast pozwala na ich przemieszczanie. Bardzo wartościowa jest tu także instrukcja COLINT, za pomocą której użytkownik może bardzo prosto obsługiwać kolizje sprite'ów — jest to forma podprogramu do którego następuje skok, gdy dany sprite zetknie się z innym bądź z podkładem. SPRCOL jest przydatna zwłaszcza wtedy, gdy nasze sprite'y projektowane są jako wielokolorowe (multicolor). RSPPOS podaje aktualne współrzędne X, Y oraz prędkość, RSPCOL informuje o przypisanych kolorach, RBUMP natomiast jakiego rodzaju kolizja nastąpiła (sprite-sprite czy sprite-podkład). Za pomocą RSPR można dowiedzieć się jakie parametry przypisano danemu sprite'owi instrukcją SPRITE.

W chwili gdy opisuję te instrukcje nie mogę się oprzeć wrażeniu, że SUPER EXPANDER był programem na którym oparto w dużej mierze ROM Commodore 128. Wrażenie to potęguje także fakt, że w firmowej instrukcji C-128 w zestawieniu skrótów poszczególnych instrukcji można spotkać np. COLINT, która w rzeczywistości nie była implementowana w tym modelu — jej odpowiednikiem jest COLLISION. Przypadek?

OBSŁUGA DŹWIĘKU

może się wydawać z pozoru nieco uboższa. Przewidziano tu jedynie instrukcje FILTER, TEMPO i TUNE umożliwiające programowanie filtrów, regulację tempa utworu oraz generalnie kształtowanie i programowanie dźwięku. Choć są to jedynie trzy instrukcje (TUNE jest tu chyba najmocniejszym punktem) możliwe jest przypisywanie jednej z 10 zaprogramowanych już obwiedni dźwięku, programowanie czasu trwania poszczególnych faz — narastania, opadania, wybrzmiewania i zaniku (ADSR), barwy tonu (instrumentu muzycznego) czy modulacji dookrężnej. W połączeniu z instrukcją FILTER daje to użytkownikowi spore możliwości z programowaniem ciekawych dźwiękowych efektów włącznie.

PIÓRO, DRĄŻKI, WIOSEŁKA i EDYTOR

są obsługiwane przez RJOY, RPEN i RPOT co umożliwia odczytywanie aktualnego położenia (stanu) wszystkich najpopularniejszych manipulatorów i czyni programowanie takiego odczytu znacznie prostszym. W połączeniu z instrukcjami graficznymi pozwala to nawet na programowanie ciekawych pod względem graficznym i muzycznym gier komputerowych. Również za bardzo przydatną należy uznać instrukcję KEY umożliwiającą przypisywanie klawiszom funkcyjnym dowolnych ciągów znaków, liczb czy instrukcji.

Niewątpliwie SUPER EXPANDER jest przydatnym programem, porównywalnym (co najmniej) ze słynnym rozszerzeniem SIMONS BASIC choć to ostatnie ma znacznie więcej instrukcji do zaoferowania. Do modułu dołączana jest oczywiście dobrze zredagowana 65-stronicowa instrukcja zawierająca dokładny opis posługiwania się poszczególnymi funkcjami i rozkazami oraz kilkanaście programów demonstrujących możliwości SUPER EXPANDERA.

Uważny Czytelnik zwrócił już zapewne uwagę, że tytuł artykułu dotyczy programu...

SUPER EXPANDER PLUS

a nie SUPER EXPANDER o którym mówiliśmy do tej pory. Nie jest to pomyłka, gdyż do testowania otrzymałem dwa moduły — jeden zawierający oryginalny program oraz drugi nazwany tak jak tytuł artykułu. W wersji drugiej program został nieco rozszerzony przez...dwójkę pomysłowych Polaków. Mowa oczywiście o tandemie Bogusław Radziszewski i Krzysztof Gajewski, autorach opisywanego już w BAJTKU 8/87 rewelacyjnego WARSAW BASIC. Wypatrzyli oni dość szybko bardzo istotną wadę SUPER EXPANDERA i postanowili ją zlikwidować, dodając niejako od siebie kilka innych, również bardzo pożytecznych możliwości.

Jak wiadomo chyba każdemu programiście, każdy dobry program graficzny powinien posiadać dwie opcje stałe: możliwość zapisania na taśmie bądź dysku danego rysunku oraz możliwość wyprowadzenia go na drukarkę. Jak na złość SUPER EXPANDER oryginalny nie ułatwiał realizacji tych zadań w żaden sposób — znów konieczne byłyby serie POKE i PEEK z czym nie każdy (zwłaszcza początkujący) programista da sobie radę. Wspominany powyżej tandem dodał więc następujące instrukcje:

@ L — umożliwia wczytanie obrazu wysokiej rozdzielczości z dyskietki bądź taśmy.

@ P — wczytuje do pamięci odpowiednią procedurę obsługującą przenoszenie naszego rysunku na drukarki MPS 803 (a więc i typy kompatybilne takie jak VIC 1525 i MPS 801), STAR NL-10 z interfejsem Commodore oraz MPS 802. Tu warto wspomnieć, że na dyskietce jaką otrzymałem wraz z instrukcją i modułem, dla MPS 802 zawarto aż 6 procedur umożliwiających wydruk obrazu o rozdzielczości 320 * 200 punktów z lewej strony kartki, na jej środku oraz dwie opisane kombinacje razem (3 procedury), 320 * 400 punktów liniami przerywanymi oraz 640 * 400 punktów (!) liniami ciągłymi bądź przerywanymi!!!

Osobiście żałuję, jedynie, że podobnych procedur nie ma dla mojej NL-10...

@ S — pozwala na zapisanie na dyskietce lub taśmie obrazu graficznego.

Wracając do procedur warto wspomnieć, że autorzy wykorzystali tu także swe doświadczenia z czasów opracowywania WARSAW BASIC. Zajmują one w pamięci ściśle określony jej obszar, tak więc wczytanie nowej procedury (powiedzmy dla drukarki NL-10) kasuje jedynie starą; ponadto jeżeli dana procedura, którą akurat wywołujemy z dyskietki jest już w pamięci, to nie będzie ona niezliczoną ilość razy wczytywana — system operacyjne „wie", że znajduje się ona już w naszej RAM.

Autorzy rozszerzenia nie poprzestali na tym. Rysunki i obrazy tworzone za pomocą SUPER EXPANDER PLUS mogą być także wczytywane i modyfikowane (w pełni, a więc do dyspozycji użytkownika są wszystkie instrukcje, także i trójwymiarowe) w WARSAW BASIC. Wymienność ta jest dwustronna — rysunki wykonane w WARSAW BASIC mogą być więc modyfikowane i odczytywane przez SUPER EXPANDER PLUS. Do rąk użytkownika trafia również estetycznie wykonana instrukcja obsługi wraz z opisem standardowych instrukcji programu jak też i tych dodanych. Nic dodać, nic ująć...

Reasumując, wydaje mi się, że należy uznać SUPER EXPANDER PLUS za program dobry, dający użytkownikowi wiele nowych i potrzebnych w Commodore 64 funkcji, co jednocześnie umożliwia wykorzystanie jego wielu ukrytych walorów, trudnych do uzyskania za pomocą oryginalnej wersji BASIC zwłaszcza dla początkujących programistów. Zalety tego programu podnoszą znacznie dodane do niego instrukcje umożliwiające zapis, odczyt i wydruk obrazów graficznych, gdyż uzupełniają one znakomicie uciążliwą lukę pozostawioną przez firmę Commodore.

Klaudiusz Dybowski

DRAGO BASIC

Chociaż Commodore 64 ma ogromne możliwości graficzne, to korzystanie z nich jest bardzo skomplikowane, zwłaszcza dla początkujących. Ze względu na dużą liczbę instrukcji POKE czy pętli FOR... NEXT programy są mało przejrzyste i trudne do zrozumienia.

Oprócz tego, niejako dodatkowo, wymagana jest od użytkownika spora wiedza dotycząca organizacji pamięci w tym komputerze, sposobów rezerwowania jej odpowiedniego obszaru dla potrzeb ekranu graficznego czy działania na bitach. Ponieważ punkt o współrzędnych 0,0 znajduje się w lewym górnym rogu ekranu tworzenie np. wykresów funkcji wymaga ponadto dodatkowych obliczeń. Z tych też powodów chciałbym przedstawić Czytelnikom „BAJTKA" rozszerzenie graficzne standardowego interpretera BASIC — DRAGO BASIC.

Istnieje już sporo rozszerzeń tego typu — GRAPHICS BASIC, SIMONS BASIC, SUPERGRAPHIC itp. Niestety możliwość korzystania z tych programów limituje w dużym stopniu brak ich polskojęzycznych opisów dość rzadko dostępnych nawet na giełdach komputerowych. Innym mankamentem jest zmniejszenie ilości dostępnej dla użytkownika pamięci — np. SIMONS BASIC zabiera jej od 8 do 16 KB, co nie pozostaje oczywiście bez wpływu na jakość naszych programów.

DRAGO BASIC nie jest rozszerzeniem w takim znaczeniu, w jakim zwykle się to rozumie. Zamiast bowiem dopisywania nowych instrukcji czy rozkazów wykorzystano tu cztery istniejące i rzadko używane instrukcje — VERIFY, WAIT, LET oraz CONT. Nowe instrukcje graficzne składają się z takiej samej ilości liter i nazywają się TRYBEK, GUMA, BAR i RYPU. Instrukcje te są „umieszczone" w tych samych miejscach pamięci ROM co stare, z tym, że zmieniono odpowiednio adresy procedur instrukcje te realizujących. Procedury te zostały umieszczone w dodatkowych 4 KB pamięci od adresu 49152 ($C000) dzięki czemu mamy do dyspozycji całe 38 KB pamięci RAM. Zaletą DRAGO BASIC jest przeniesienie punktu o współrzędnych 0,0 w lewy DOLNY róg ekranu czyli tak jak na osi współrzędnych. Program ten pozwala nam na tworzenie grafiki w standardowym trybie graficznym Commodore — rozdzielczość 320∗200 punktów, dwa kolory.

Podczas wpisywania DRAGO BASIC należy zwrócić baczną uwagę, aby nie pomylić się zwłaszcza podczas wpisywania linii z DATA — liczby zawarte w tych liniach to właśnie nasze nowe procedury. Ponieważ po wpisaniu kodu maszynowego program samoczynnie się kasuje, ważne jest również, aby przed jego uruchomieniem zapisać go najpierw na taśmie czy dyskietce. Jeżeli program zostanie wpisany poprawnie, po uruchomieniu pojawi się na ekranie plansza tytułowa oraz usłyszymy melodię — program jest gotowy do użytku.

Nowe instrukcje mają następującą postać:

TRYBEK (parametr)

Instrukcja ta służy do zmiany ekranu graficznego na tekstowy i odwrotnie. TRYBEK 1 pozwala nam na przejście do trybu pracy graficznego, TRYBEK 0 z kolei umożliwia powrót do trybu tekstowego. Gdy parametr nie będzie równy ani 1 ani 0, komputer wyświetli komunikat ILLEGAL QUANTITY ERROR.

GUMA

Jest to odpowiednik PRINT CHR$(147) w trybie tekstowym — GUMA pozwala nam skasować (bezpowrotnie) całą zawartość ekranu graficznego.

BAR kolor kreski, kolor tła

Za pomocą BAR możliwe jest przypisanie wybranych kolorów dla kreski (punktu) oraz tła ekranu graficznego. Numery kolorów są zgodne z numerami podanymi w instrukcji obsługi komputera (np. 0 = czarny, 1 = biały itd.).

RYPU X,Y

RYPU umożliwia nam wykreślenie punktu na ekranie graficznym. Parametr X może przybierać wartości w zakresie 0-319, parametr Y w zakresie 0-199. Należy pamiętać, że punkt o współrzędnych 0,0 znajduje się w DOLNYM lewym rogu ekranu. Powrót do normalnego trybu pracy komputera możemy uzyskać za pomocą kombinacji klawiszy STOP i RESTORE wciśniętych jednocześnie. Powrót do programu jest możliwy za pomocą POKE 1,54. Symbol „(CLR)" w programie oznacza czyszczenie ekranu.

A oto przykładowy program pozwalający nam na wykreślenie paraboli:

```
10 TRYBEK 1 : GUMA 1,0 : BAR
   1,0
20 FOR X=0 TO 240 : Y=.01∗(X−
   120) ↑ 2+20
30 RYPU X,Y : NEXT
40 FOR X=0 TO 240 : RYPU X,20 :
   NEXT
50 FOR Y=0 TO 199 : RYPU 120,Y
   : NEXT
60 GET A$ : IF A$="" THEN 60
70 TRYBEK 0 : END
```

Jan Jasiński
„COMMODORE CLAN KOMODA"

```
100 REM ****** DRAGO BASIC V1.1 ******
101 :
110 PRINT"(CLR)":POKE53280,0:POKE53281,0
120 POKE646,1:PRINT"PROSZE CZEKAC 30 SEK..."
130 A=0:REM * INICJALIZACJA SUMY KONTROLNEJ *
140 REM *** PRZENOSZENIE ROM BASIC DO RAM ***
150 FORI=40960TO49151:POKEI,PEEK(I):NEXT
160 REM *** ZMIANA LET NA BAR ***
170 FORI=41150TO41152:READN:POKEI,N:A=A+N:NEXT
180 READL,H:POKE40988,L:POKE40989,H:A=A+L+H
190 DATA 066,065,210,075,196
200 REM *** ZMIANA WAIT NA RYPU ***
210 FORI=41189TO41192:READN:POKEI,N:A=A+N:NEXT
220 READL,H:POKE41008,L:POKE41009,H:A=A+L+H
230 DATA 082,089,080,213,130,196
240 REM *** ZMIANA CONT NA GUMA ***
250 FORI=41225TO41228:READN:POKEI,N:A=A+N:NEXT
260 READL,H:POKE41024,L:POKE41025,H:A=A+L+H
270 DATA 071,085,077,193,053,196
280 REM *** ZMIANA VERIFY NA TRYBEK ***
290 FORI=41201TO41206:READN:POKEI,N:A=A+N:NEXT
300 READL,H:POKE41014,L:POKE41015,H:A=A+L+H
310 DATA 084,082,089,066,069,203,11,196
320 REM *** REM ZMIANA KOMUNIKATU BLEDU ***
330 FORI=42042TO42044:READN:POKEI,N:A=A+N:NEXT
340 DATA 076,000,196
350 REM *** WCZYTYWANIE NOWYCH PROCEDUR ***
360 FORI=50176TO50480:READN:POKEI,N:A=A+N:NEXT
370 IFA<>39050THENPRINT"NIEDOBRE DANE '":END
380 PRINT"(CLR)      *** DRAGO BASIC V1.1  ***"
390 PRINT"                    ,"
400 PRINT"  (C)  1987 KRAKOW, TEL. 11-07-34"
410 PRINT:PRINT"          38911 BYTES FREE"
420 GOSUB 780
430 POKE 1,54:NEW
440 DATA 032,024,196,138,010,170,076,061,164
450 DATA 080,070,083,032.158.183,224,001,144
460 DATA 005,240,019,076,072,178,169,027,141
470 DATA 017.208,169,021,141,024.208,169,151
480 DATA 141,000,221,096,169,059,141,017,208
490 DATA 169,008,141,024,208,169,148,208,238
500 DATA 162,032,169,224,133,252,160,000,132
510 DATA 251,152,145,251,200,208,251,230,252
520 DATA 202,208,246,096,032,123,196,138,010
530 DATA 010,010,010,133,002,032,253,174,032
540 DATA 123,196,138,005,002,160,192,132,252
550 DATA 160,000,132,251,162,002,145,251.200
560 DATA 208,251,230,252,202,016,246,145,251
570 DATA 200,192,232,144,249,096,032,158,183
580 DATA 224,016,176,017,096,032,235,183,134
590 DATA 002,169,199,056,229,002,133,002,201
600 DATA 200,144,003,076,072,178,165,021,240
610 DATA 010,201.001,208,245,165,020,201,064
620 DATA 176,239,169,000,133,251,169,224,133
630 DATA 252,165,020,041,248,024,101,251,133
640 DATA 251,165,021,101,252,133,252,165,002
650 DATA 041,007,024.101,251,133,251.144.002
660 DATA 230,252,165,002,074,074,074,010,170
670 DATA 189,247,196,024,101,251,133.251,189
680 DATA 248.196,101,252.133,252,165.020,041
690 DATA 007,170,160,000.120.169,052,133,001
700 DATA 177,251,029,041,197,145,251,169,054
710 DATA 133,001,088,096.000,000,064.001,128
720 DATA 002,192,003,000,005,064,006,128,007
730 DATA 192,008.000,010.064,011,128.012,192
740 DATA 013,000.015,064,016,128,017.192,018
750 DATA 000,020,064,021.128,022,192,023,000
760 DATA 025,064,026,128,027,192,028,000,030
770 DATA 128,064,032,016,008,004,002.001
780 B=54272
785 FORCL=BTOCL+24:POKECL,0:NEXT:POKEB+5,85
790 POKEB+6,85:POKEB+12,85:POKEB+13,85
800 POKEB+24,15:POKEB+4.33:POKEB+11.17
810 FORX=1TO6:READH1,L1,H2,L2:POKEB+1,H1
820 POKEB,L1:POKEB+8,H2:POKEB+7,L2
830 IFH1=50THENFORT=1TO200:NEXT
840 FORT=1TO100:NEXT
850 DATA 025,030,018,209,033,135,025,030,042
860 DATA 062,031,165,050.060,037,162,042,062
870 DATA 031,165,050,060,037,162
880 NEXT:POKEB+4,32:POKEB+11,16
890 FORW=1TO500:NEXT
900 FORCL=BTOB+24:POKECL,0:NEXT:RETURN
```

„PCHEŁKA"

Mianem tym opatrywane są często programiki, które z reguły zawierają się w kilkunastu, a najwyżej kilkudziesięciu bajtach.

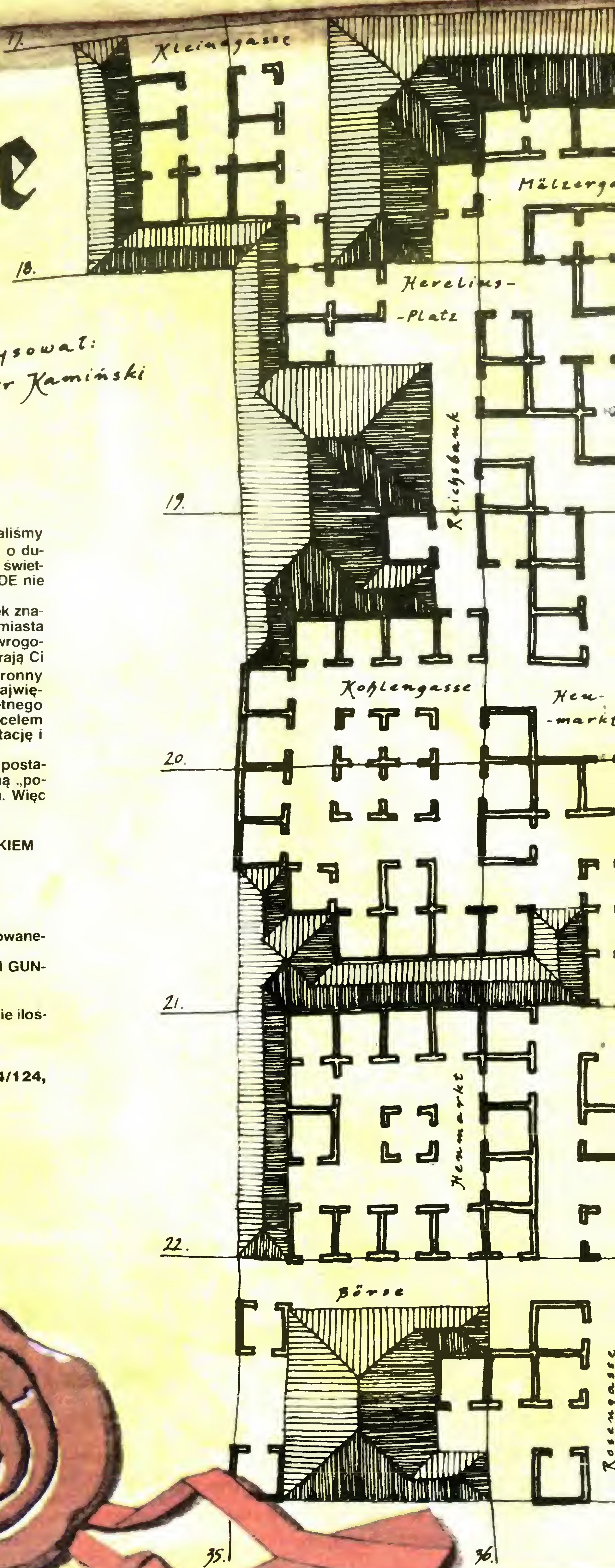
Poniższy program (dla C-64) może być z powodzeniem wykorzystywany do dobierania odpowiedniego koloru ekranu, tekstu bądź ramki za pomocą wciśnięcia klawisza RESTORE. Co ciekawsze, sam program zajmuje dokładnie ... 6 bajtów pamięci, pozostałych 12 służy do zmiany wektora procedury przerwania NMI. Po zmianie linii 70 na **DATA 238, 032, 208** wciśnięcie RESTORE spowoduje zmiany koloru ramki, natomiast **DATA 238, 134, 002** zmiany koloru tekstu. Programik ten może być z powodzeniem stosowany w innych „dużych" programach.

(kd)

```
10 rem *** pchełka $00 ***
15 :
20 rem     k. dybowski
25 :
30 for i=680 to 698 : read q
35 poke i,q : next
40 sys 680 : new
45 :
50 data 120, 169, 181, 141
55 data 024, 003, 169, 002
60 data 141, 025, 088, 096
65 :
70 data 238, 033, 208
75 data 076, 071, 254
```

night shade city

Rysował: Piotr Kamiński



NIGHT SHADE to następna gra firmy Ultimate. Prezentowaliśmy już w Bajtku: ALIEN8, GUNFRIGHT, PENTAGRAM, teraz coś o duchach, wampirach, kościotrupach. Firam ULTIMATE słynie ze świetnej oprawy graficznej w swoich programach a i NIGHT SHADE nie odbiega zbytnio od „normy".

Bohaterem gry jest młody podróżnik, który przez przypadek znalazł się w podziemnym „Mieście Zmarłych" a wydostać się z miasta to nie lada problem. Na każdym kroku czyhają tu śmiertelni wrogowie, stwory bez „ducha" i „ciała", które po dotknięciu zabierają Ci pewien procent twoich „sił witalnych". Ale nie jesteś bezbronny — broń na te stwory musisz sam odnaleźć i zebrać jej jak najwięcej, (diagram pokazuje jaka broń jest skuteczna na konkretnego stwora). Lecz zabijanie co pomniejszych stworów nie jest celem (a trafienie takiego stwora złą „bronią" powoduje jego mutację i to że staje się „nieśmiertelny").

W tym labiryncie ulic musisz odnaleźć, po kolei, cztery „postacie", znaleźć broń na nie, a następnie zabić je. Ale konkretną „postać" można zabić tylko przeznaczoną, do tego celu, bronią. Więc która broń jest skuteczna i na kogo?

Otóż:

1) SZKIELET — zabijasz go swego rodzaju MŁOTKIEM
2) ŚMIERĆ — zabijasz KLEPSYDRĄ
3) WAMPIRA — KRZYŻEM
4) DUCHA — KSIĄŻKĄ

Nie wszystko, jednak, co się rusza można zabić.

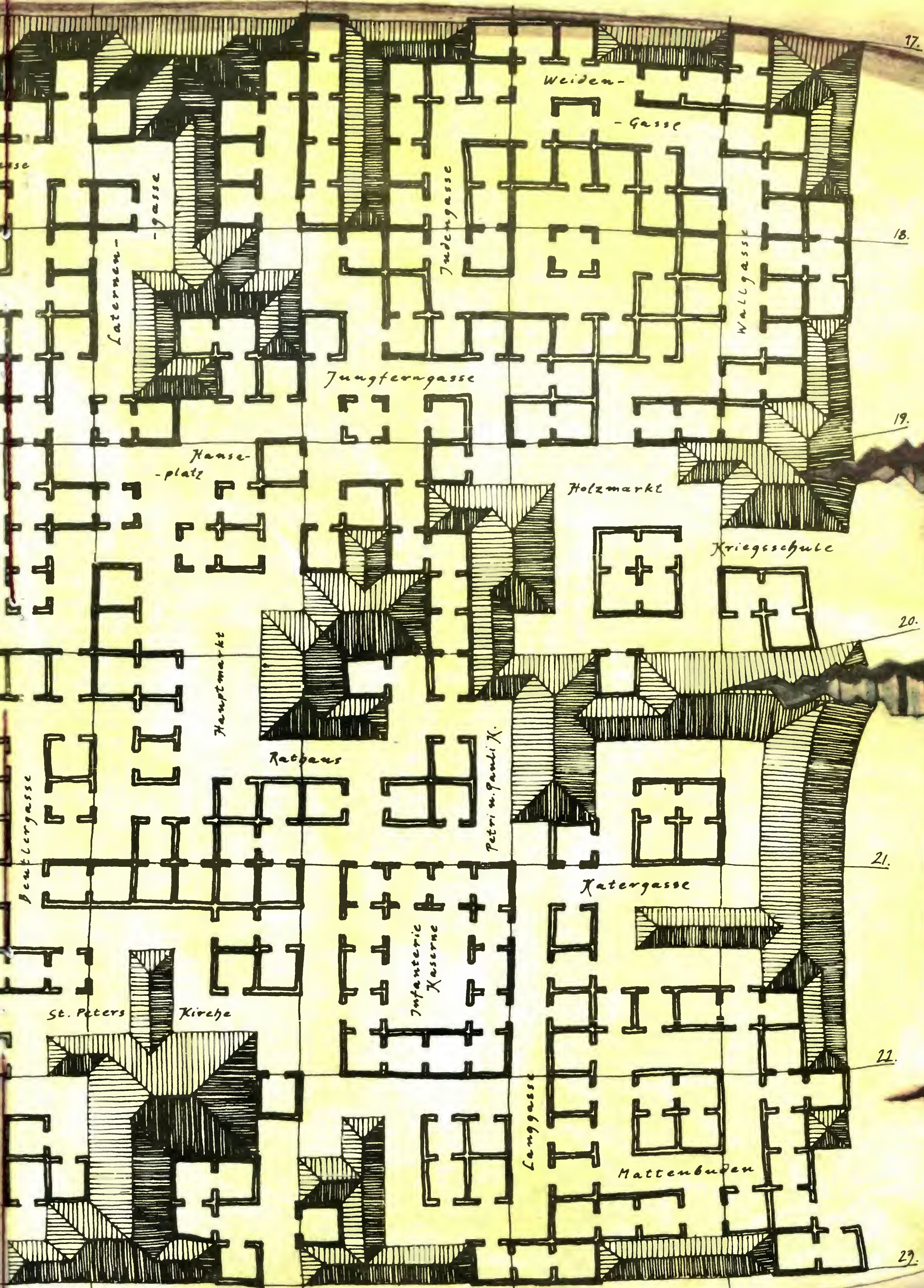
Uwaga na „RUCHOMY OGIEŃ" i „OBŁOCZEK" czy „zmutowanego stwora".

Po nabraniu pewnej wprawy gra jest trochę trudniejsza od GUNFRIGHT'a.

Zaczynamy więc grę.

W tabeli wykaz broni i stworów w mieście. 2 — podwojenie ilości stworów po strzale z broni.

GRA: NIGHT SHADE
KOMPUTER: AMSTRAD/SCHNEIDER, COMMODORE 64/124, ZX SPECTRUM 48/+/128/+2/+3
FIRMA: ULTIMATE.


17.
Weiden-
-Gasse
Laternen-
-gasse
Judengasse
18.
Wallgasse
Jungferngasse
19.
Hanse-
-platz
Holzmarkt
Kriegsschule
20.
Hauptmarkt
Rathaus
Petri u. Pauli K.
Beutlergasse
21.
Katergasse
Infanterie
Kaserne
St. Peters
Kirche
22.
Langgasse
Mattenbuden
23.
37.
38.
39.
40.

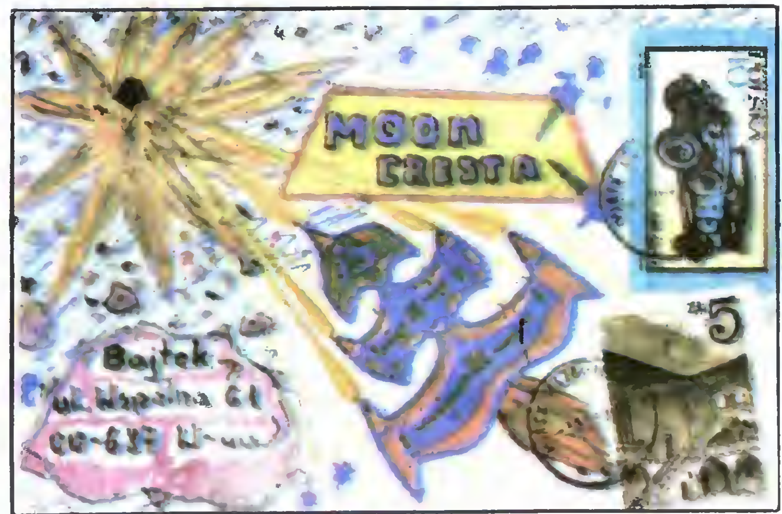
10 BAJTKOWA LISTA PRZEBOJÓW (12/87)

Chociaż na naszej liście wciąż pojawiają się nowe tytuły, to jednak starsze pozycje trzymają się mocno. Niektóre gry jak bumerang wracają na czołowe miejsca. Na dwunaste tegoroczne notowanie napłynęło 2528 propozycji. Czytelnicy głosowali na 174 tytuły gier.

			ATARI	AMSTRAD	COMMODORE	SPECTRUM
1	MISS PACMAN	↑	×	×	×	×
2	GREAT ESCAPE	↑		×	×	×
3	TRAP DOOR	↓		×	×	×
4	BARBARIAN	↑	×	×	×	×
5	WIZARD'S LAIR	↓		×	×	×
6	REVOLUTION	↓		×		×
7	XENO	↑	×		×	×
8	WINTER GAMES	↓	×	×	×	×
9	TAU CETI	↓		×	×	×
10	GLADIATOR	!		×	×	×

Nagrody-zestawy programów komputerowych — otrzymują: Alicja Mrozińska i Maciej Kurzajewski.

Sławek



Kopertę nadesłał Bartosz Mazarka z Mińska Mazowieckiego ul. Obrońców Stalingradu 25/4

ZŁOTA DZIESIĄTKA ROKU 1987

Tak, jak przed rokiem zestawiliśmy notowania Bajtkowej Listy Przebojów. Za pierwsze miejsce przyznaliśmy 10, za ostatnie 1 punkt. Oto, co się okazało:

1 BOULDER DASH — First Star Software.

Pierwsza część przygód Rockforda w podziemnych jaskiniach długo nie opuszczała Listy Przebojów. Wymaga potężnego skupienia się i maksymalnego refleksu; opieszali dostaną kamieniem po głowie.

2 SPY VS SPY II — First Star Software.

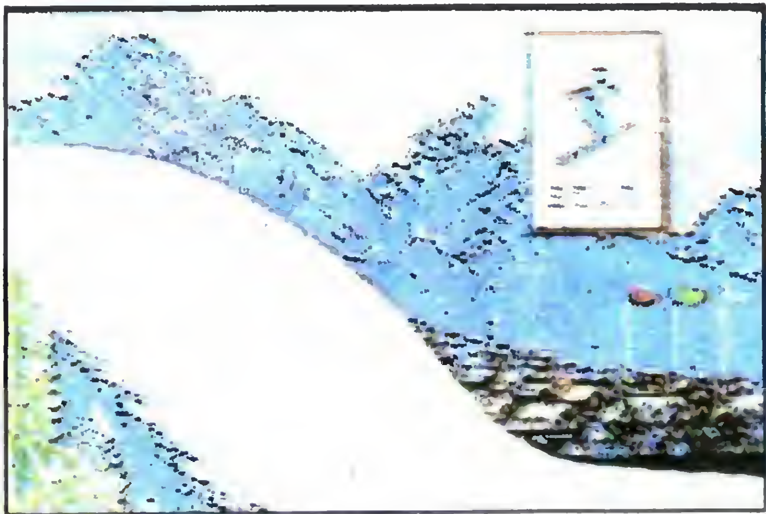
Tu już trzeba pomyśleć. Jak wykiwać nieprzyjaznego szpiega-rywala? — można np. wykopać dołek. Ale kto kopie dołki, ten sam w nie wpada (i z reguły trudno jest mu się wydostać). Ta część walki dwóch szpiegów, rozgrywająca się na wulkanicznej wysepce jest daleko trudniejsza od pierwszej.

3 BROADSIDES — S.S.I.

Abordaż w epoce atomowych okrętów podwodnych? Ależ tak! Radzę jednak najpierw wypróbować kartacze (nazwa Broadsides mówi sama za siebie). Akcja rozgrywa się wolno, ale kto wymaga zwrotności od XVI-wiecznych karawel?

4 WINTER GAMES — US GOLD.

Jedna z długiej listy gier sportowych — o dziwo — nie wymaga machania joystickiem w lewo i w prawo bez ustanku. W dwóch częściach tylko 7 konkurencji, ale proszę zwrócić uwagę na wspaniałe alpejskie pejzaże i muzykę



5 WIZARD'S LAIR — Bubble Bus Software.

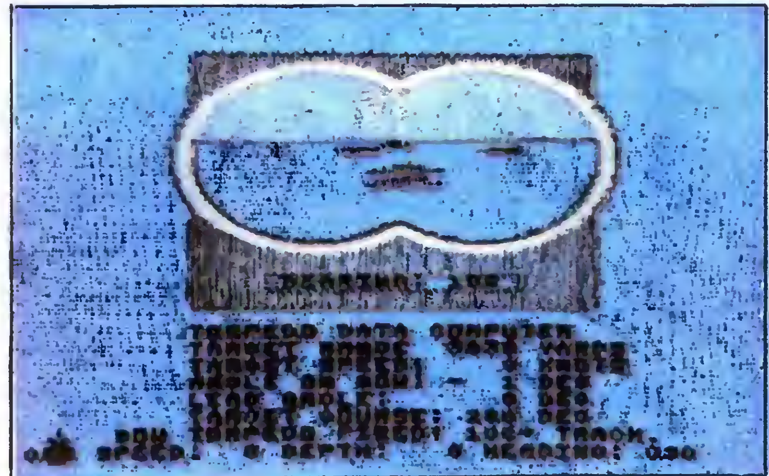
Pomysł jest stary — wydostać się z labiryntu wykonując określone zadania. Gra jest niejako połączeniem gier SABRE WULF i ATIC ATAC. Świetna grafika i (jak zwykle) melodyjne wstawki na pewno uatrakcyjniają zabawę.

6 BEACH HEAD II — US GOLD.

Po dobiciu konwojem do brzegu i zdobyciu bunkra (W BEACH HEAD) rozpoczyna się marsz w głąb lądu. Wieńczy go pojedynek z Dyktatorem. Gra jest niewątpliwie lepsza od pierwszej części, akcja jest szybka, a grać można jako atakujący, broniący lub we dwóch z kolegą.

7 SILENT SERVICE — Micro Prose Software.

Coś dla koneserów. Doskonała symulacja okrętu podwodnego rozgrywająca się podczas II wojny światowej (1942–1944 r.) na Pacyfiku. Do wyboru trzy opcje gry, w WAR PATROLS jeden z pięciu patroli wojennych.



8 TRAP DOOR — Macmillian Software.

Jak w każdej grze tej firmy, bohater ma wysokość 1/2 ekranu i jest bardzo starannie animowany. Wymagana jest zdolność szybkiego kojarzenia faktów, zręczność też się przyda



9 DAN DARE — Gang Of Five.

Gra także labiryntowa, o uproszczonym scenariuszu. Jednak ratowanie Ziemi przed zagładą jest niewątpliwie zadaniem odpowiedzialnym i widok rozpadającej się w drobny mak wrogiej planety będzie miły oku każdego gracza.

10 URIDIUM — Hewson.

Gra typowo „automatowa". Jeżeli posiada jakiś sens to bardzo głęboko ukryty. Chociaż latanie i strzelanie nad kosmicznym krążownikiem nie jest proste (palec bardzo drętwieje), to Czytelnicy dwukrotnie umieścili ją na pierwszym miejscu.

Dla ciekawskich podajemy też drugą dziesiątkę:

11. CAULDRON
12. GREAT ESCAPE
13. W. A. R.
14. TIGERS IN THE SNOW
15. SPY VS SPY
16. WARHAWK
17. REVOLUTION
18. FIRE LORD
19. SEVEN CITIES OF GOLS
20. TAU CETI



CHEQUERED FLAG

„... **umieszcza cię** za kierownicą. Będziesz musiał patrzyć jednym okiem na tablicę przyrządów, operować biegami i hamować by uniknąć ryzyka. Wszystko to w poszukiwaniu rekordu". Oto fragment tekstu zamieszczonego przez firmę PSION na etykiecie tej gry. Skąd my to znamy? ROAD RACE, FORMULA 1, POLE POSITION itd. Na co komu jeszcze jedno? I tu zaczyna się przetarg, bowiem każda z tych gier jest na swój sposób wspaniała: POLE POSITION — staranna grafika, wyścig poprzedzony tzw. „oficjalnym treningiem", możliwość premiowanej jazdy, lista rekordów: ROAD RACE — poza ciekawą grafiką gra opatrzona pewną fabułą, duża różnorodność warunków jazdy, pozorowana zmiana biegów; FORMULA 1 — możliwość wyboru toru, stanu nawierzchni, możliwość jazdy z automatyczną skrzynią biegów, bardziej skomplikowana zmiana biegów. CHEQUERED FLAG jest grą zbliżoną w swej budowie do FORMULA 1. Można jednak w niej znaleźć kilka nowych, ciekawych rozwiązań. Każdy z dziesięciu torów opatrzony jest własnym, zapamiętanym rekordem. Gracz wybiera ponadto ilość okrążeń (do 99). Ale największą ciekawostką jest możliwość wybrania samochodu, który chce się prowadzić. Ma to bardzo duże znaczenie ze względu na różnice pomiędzy tymi fikcyjnymi maszynami. Postaram się więc krótko je przedstawić:

McFASTER SPECIAL — wyposażony w automatyczną skrzynię biegów, jest najbardziej odpowiednim samochodem dla niedoświadczonych „kierowców". Tym niemniej pozostaje szybkim (500 bhp) i niezawodnym pojazdem.

PSION PEGASUS — osiąga 560 bhp między 5000 a 10000 obrotów i cechuje się wyjątkowo niskim współczynnikiem oporu powietrza. To bardzo szybki samochód, a kierowanie nim wymaga umiarkowanych umiejętności.

FERETTI TURBO — bardzo mocny, wyposażony w turbodoładowanie. Osiąga 640 bhp między 8000 a 10000 obrotów. Bardzo trudny do prowadzenia, wskazany raczej dla doświadczonych „kierowców".

Teoretycznie gracz jest zmuszony zaangażować wszystkie dziesięć palców, by mieć kontrolę nad każdym mechanizmem. Musi bowiem obsłużyć: przyspieszacz — Q, hamulce — I, zmianę biegów na niższy — N (lub dowolny klawisz w lewo od N), zmianę biegu na wyższy — M (lub dowolny klawisz w prawo od M), szybki skręt w lewo — A, powolny skręt w lewo — S, powolny skręt w prawo — D, szybki skręt w prawo — F, zatrzymanie gry — H (kontynuacja przez ponowne naciśnięcie) oraz rozpoczęcie gry od nowa — H i T jednocześnie. Jeśli chodzi o tor, to nawet gdy wybierze Monaco czy Silverstone, będzie musiał uważać na plamy oleju, kałuże wody, rozbite szkło i kamienie. W przypadku dłuższej jazdy należy skorzystać z „pit stopu", by uzupełnić paliwo.

Gra zrobiona jest na tyle dokładnie, że po kraksie gracz powiadamiany jest o jej przyczynach, zaś po ukończeniu własnego wyścigu z czasem wie, ile godzin będą zmuszeni zużyć mechanicy z jego „stajni" na dokonanie koniecznych napraw.

Drodzy gracze! Nie dajcie się zwieść zapisom jakie zastaniecie przy poszczególnych trasach! Możecie uznać się koneserami CHEQUERED FLAG, jeśli na większości torów uzyskacie wyniki poniżej 1 min. 30 sek.

Komputer: ZX Spectrum 48/+, Commodore 64, Amstrad/Schneider 464/6128

(pb)

S.O.S.

Jak uzyskać nieśmiertelność w grach: STRIP POKER, GHOST'N GOBLINS, HIGWAY ENCOUNTER w wersji na Amstrada 464?

Tomasz Herbst
ul. Świerczewskiego 12
19-500 Gołdap

Mam kłopot w grze DAN DARE w wersji ZX Spectrum. Jestem na pierwszym poziomie i nie potrafię zjechać windą na dół.

Artur Bujniewicz
ul. Mickiewicza 6/3
78-520 Złocieniec

Poszukuję instrukcji do gry INDIANA JOE na ZX Spectrum.

Maciej Odoj
ul. Sobieskiego 10
42-730 Korzęcin

Jestem użytkownikiem CPC 464. Bardzo lubię grać w grę SABOTEUR firmy Durell lecz nie wiem do czego służą deski i młotki oraz jak je zebrać. Pomóżcie!

Michał Wieczorek
ul. Leszczyńska 6/10
41-806 Zabrze

Poszukuję dokładnego opisu do gry ZORRO na Atari 800XL.

Agnieszka Sielicka
ul. Raduńska
83-333 Chmielno

Poszukuję POKE'ów do gier JET MEN, HARIER, EL-STORM, GRAND ATACK w wersji na ZX Spectrum. Chętnie wymienię je z innym POKErzystą.

Grzegorz Pawlak
ul. Skalskiego 3/34
42-505 Będzin

Jestem użytkownikiem Atari 800XL. Potrzebuję pomocy w grach: E.T., DIMENSION X, BUCK ROGERS i SPY Vs SPY.

Marcin Budyn
ul. Sportowa 133
39-200 Dębica

Proszę o pomoc w następujących grach: BRUCE LEE ROAD RACE COSMIC TUNELS w wersji na Atari 130 XE

Tomasz Adamski
ul. Hubala 4/2
94-042 Łódź

Poszukuję kodu do gry RAMBO na Atari 800XL.

Paweł Kuchnik
ul. Jordana 8/79
32-500 Chrzanów

Proszę o pomoc w grach na Commodore 64: W.A.R., SPY Vs SPY III THE LAST NINJA (co zrobić z ziejącym ogniem smokiem).

Marek Ptaszyński
ul. Lumumby 101/105
80-371 Gdańsk

Mam 12 lat. Ponieważ słabo poruszam się o kulach, większość czasu spędzam w domu, również uczę się indywidualnie w domu. Rodzice długo odkładali pieniądze, aby spełnić moje marzenia. I właśnie od września br. jestem szczęśliwym posiadaczem mikrokomputera „Atari 130 XE". Mimo, że posiadam kilka gier, to trzech z nich nie mogę do końca rozegrać. Chodzi mi o gry:

1. „MINER" 2049 — nie mogę przejść z trzeciego etapu (wydaje mi się, że należy uzyskać nieśmiertelność Minera)
2. PITFAL II — tutaj również należy
3. NEPTUN'S DAUGHTER'S — uzyskać nieśmiertelność

Może ktoś z czytelników pomoże mi.

Wiadomo też, że gry na „Atari" są drogie dlatego proszę i wierzę, że znajdzie się ktoś życzliwy z „Bajtkowych" Czytelników, który udostępni mi ciekawe gry, a może i programy. Będę za to bardzo wdzięczny.

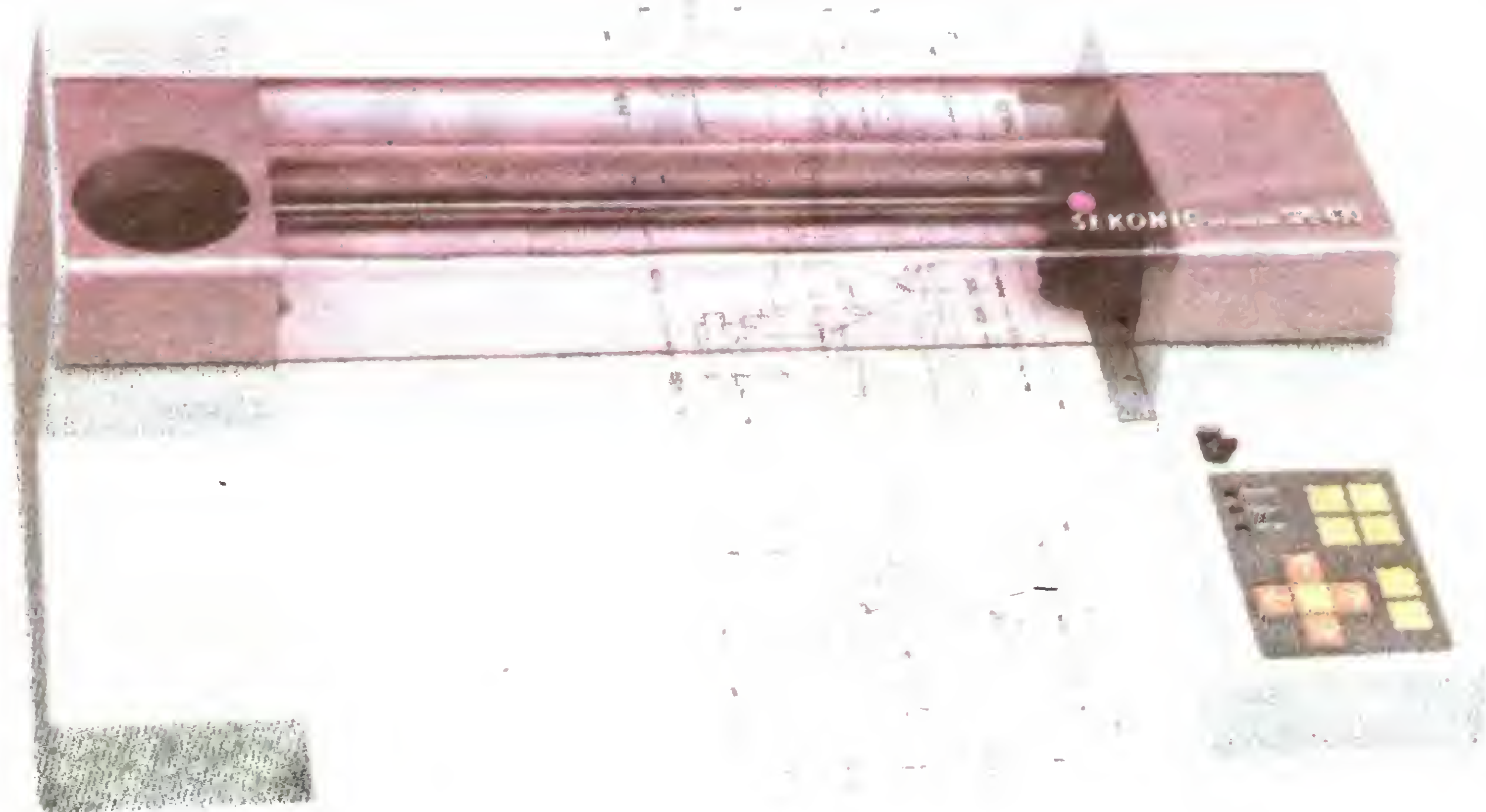
Arkadiusz Skrzypiński
70-463 Szczecin
ul. Mariana Buczka 24A/12

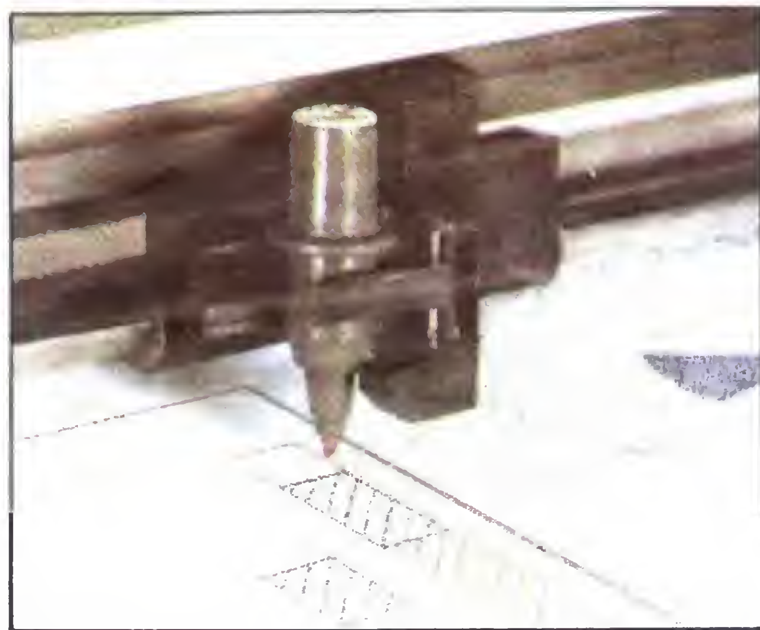
Od redakcji: Przesyłamy Arkadiuszowi dwie kasety z programami na Atari. Liczymy, że Czytelnicy „Bajtka" również mu pomogą.

KRÓL I KRÓLOWA GIER

Alicja Mrozińska, lat 13, uczennica VII klasy Szkoły Podstawowej Posiadany (i ulubiony) mikrokomputer to Commodore, a jej ulubiona gra to LAZY JONES.

Maciej Kurzajewski, lat 14, uczeń Szkoły Podstawowej nr 3 w Kaliszu. Posiada komputer Sharp MZ-700 a jego ulubiona gra to JACK THE NIPPER. Prócz komputerów posiada jeszcze jedną pasję, jest nią muzyka i śpiew.





Wysokiej jakości rysunki wykonuje się na ploterach

KOMPUTER
W BIURZE KONSTRUKCYJNYM

CAD to skrót angielskiego terminu „Computer Aided Design", co w języku polskim oznacza komputerowo wspomagane projektowanie.

CAD

Na początek trochę historii rozwoju CAD. W 1948 roku na uniwersytecie Manchester w Wielkiej Brytanii zostaje uruchomiony pierwszy program zapisany w pamięci komputera Mark I. Pierwsze prace nad systemami CAD podjęto w Stanach Zjednoczonych w kilku wielkich przedsiębiorstwach przemysłu lotniczego i motoryzacyjnego. Na początku lat pięćdziesiątych takie prace podjęto w koncernie Mc Donell — Douglas. Ówczesne komputery były zastosowane do wykonywania obliczeń numerycznych. Prowadzone obliczenia sprawdzające dla konstrukcji bardzo drogich i bardzo odpowiedzialnych, których koszt znacznie przewyższał koszt samych komputerów.

Z biegiem lat bardzo szybko rozwijały się takie dziedziny nauki jak elektronika, mechanika analityczna, wytrzymałość materiałów, oraz różne metody zastosowań matematyki. Przyszły lata siedemdziesiąte a wraz z nimi era mikroprocesorów. W 1972 roku wyprodukowano mikroprocesor 8-bitowy INTEL 8008, a w 1978 roku mikroprocesor 16-bitowy INTEL 8086. Dla systemów CAD wprowadzenie mikroprocesorów 16-bitowych miało przełomowe znaczenie, ponieważ dopiero one posiadały wystarczającą moc obliczeniową oraz możliwość adresowania odpowiedniej ilości pamięci. Były przy tym dosyć tanie i można je było powszechnie zastosować. Pojawiły się szybkie pamięci masowe oraz monitory graficzne o wysokiej rozdzielczości, nierzadko 1024 × 1024 punkty.

Obecnie na świecie pracuje kilkadziesiąt tysięcy systemów CAD i oferuje się co najmniej kilkadziesiąt różnych jego wersji. Praktycznie wszystkie pozwalają na wykonywanie rysunków technicznych na ekranie monitora. 80% tych systemów umożliwia dwuwymiarową, a pozostałe 20% trójwymiarową prezentację graficzną elementów konstrukcyjnych.

Systemy CAD, to również pakiety programów do prowadzenia obliczeń wytrzymałościowych, optymalizacji konstrukcji, budowy modeli symulacyjnych, oraz duże bazy danych z katalogami rozwiązań technicznych, katalogami części znirmalizowanych i powtarzalnych. Pakiety programów do prowadzenia obliczeń są zwykle wykonywane przez użytkowników, ponieważ każdy zakład ma własne metody prowadzenia obliczeń związane z ich specjalizacją.

W 1984 roku wyprodukowano mikroprocesor 32-bitowy Motorola 68020 posiadający bardzo dużą szybkość przetwarzania oraz bardzo duże możliwości adresowania pamięci. Stosując mikroprocesory 32-bitowe buduje się superminikomputery, które pozwalają na zbudowanie kilkudziesięciu sprzężonych stanowisk CAD.

Korzyści płynące z zastosowania systemów CAD są następujące:
— znaczne skrócenie czasu wykonania dokumentacji przy mniejszej liczbie błędów,
— zautomatyzowanie prac kreślarskich,
— odciążenie projektanta od pracy rutynowych, nietwórczych,
— możliwość zwiększania poziomu szczegółowości modelu projektowanego obiektu, wpływająca często na jakość wyrobu,
— polepszenie jakości rozwiązań projektowych uzyskiwane dzięki możliwości tworzenia zbiorów dopuszczalnych rozwiązań i ich optymalizacji,
— wykorzystanie danych geometrycznych dla programowania obrabiarek sterowanych numerycznie (NC).

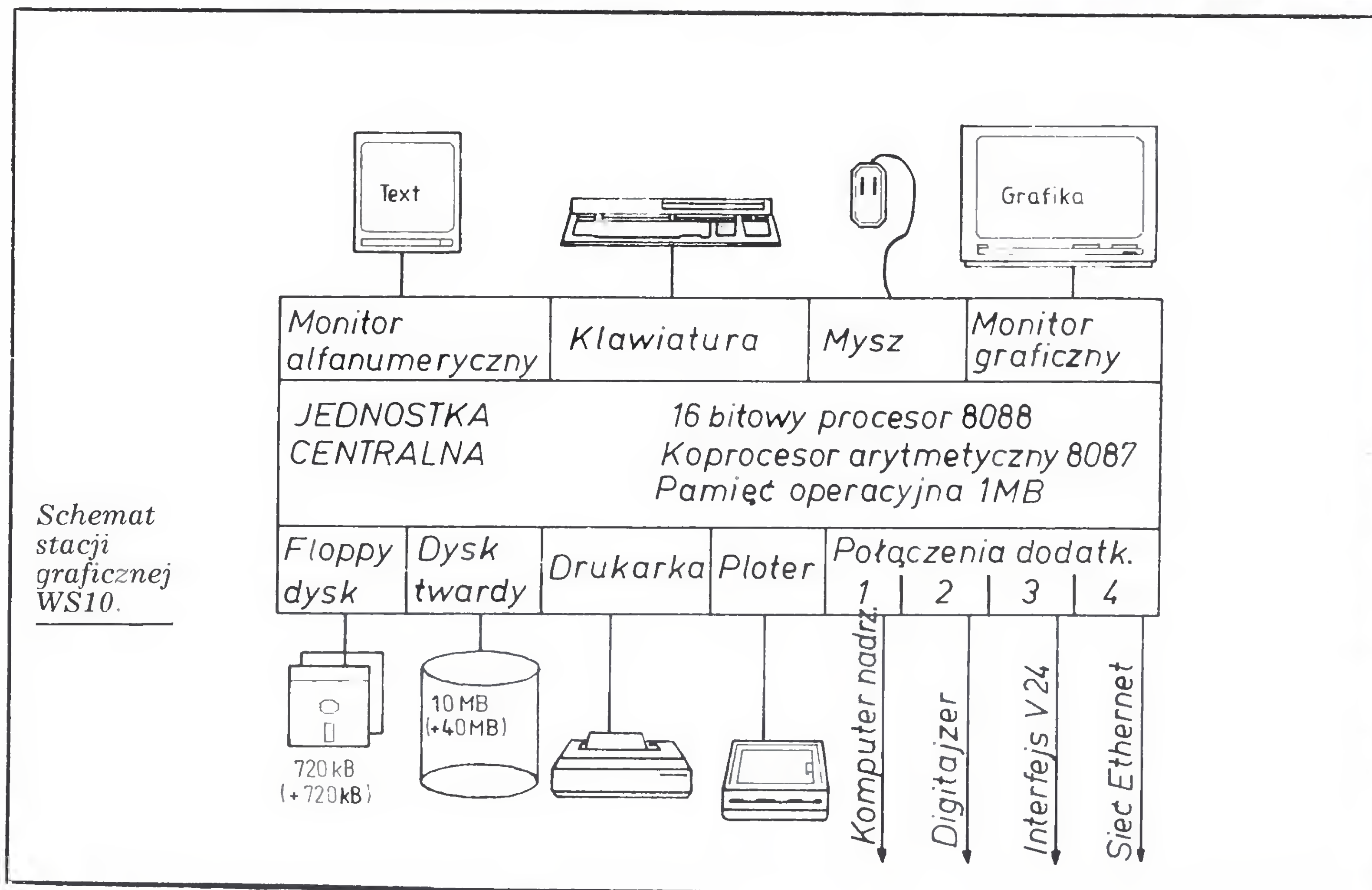
SPRZĘT

W ostatnich latach na rynku pojawiło się wiele mikrokomputerowych systemów CAD różnej wielkości i o różnych możliwościach, a co za tym idzie zróżnicowanej cenie. Producenci tych systemów dostarczają oprogramowanie specjalizowane dla potrzeb użytkownika lub otwarte, pozostawiając użytkownikowi możliwość wprowadzania modyfikacji i rozszerzeń. Obecnie dominuje tendencja sprzedawania systemów samodzielnych stanowisk CAD, podczas gdy w poprzednich latach preferowano bardziej rozbudowane i kosztowniejsze systemy komputerowe.

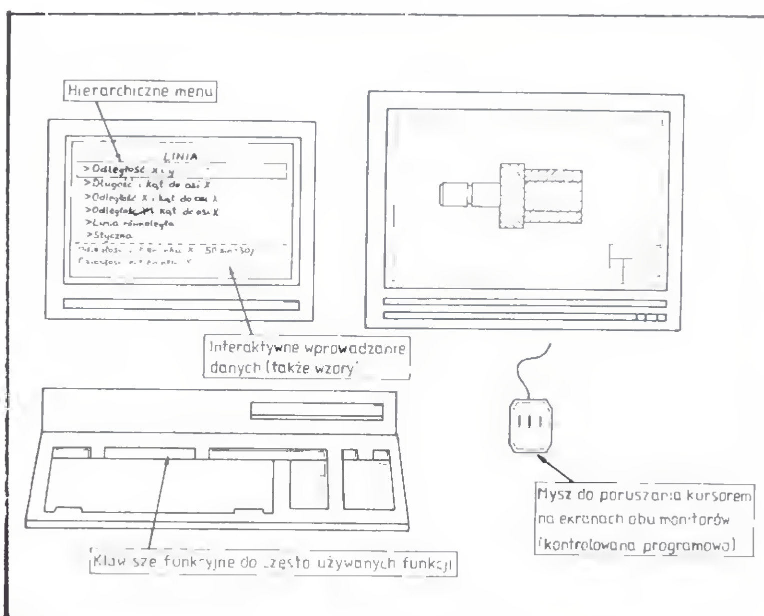
Przykładem systemu spełniającego wszystkie wymogi autonomicznego stanowiska komputerowego wspomagania prac projektowych jest stacja graficzna WS10 firmy SIEMENS. Jej cechy użytkowe sprawiają, że jest ona szczególnie przydatna dla małych i średniej wielkości biur projektowych. Jednostkę centralną stanowi produkt koncernu SIEMENSA, komputer SICOMP PC wyposażony w 16-bitowy mikroprocesor INTER 8088, wspomagany przez koprocesor arytmetyczny 8087. Dalsze jego cechy użytkowe to: pamięć operacyjna 1MB, pamięć na dysku twardym pojemności 10 MB typu Winchester, stacja dysków elastycznych 720 KB z możliwością podłączenia drugiej bliźniaczej, monitor alfanumeryczny 12-calowy z interakcyjnym ekranem, monitor graficzny 14-calowy o 8 kolorach i wysokiej rozdzielczości 640 × 480 punktów.
Stanowisko posiada pulpit graficzny do wprowadzania poleceń przy użyciu myszy.

Przesuwając mysz powodujemy przemieszczanie po ekranie kursora oznaczającego dany punkt, a wciskając odpowiedni przycisk wprowadzamy automatycznie współrzędne do komputera. W zestawie znajdują się interface'y do podłączenia standardowej lub kolorowej drukarki, dzięki której projektant może na bieżąco kopiować obraz z monitora graficznego. Gotowe, wysokiej jakości rysunki wykonuje się na ploterach w formacie A4, A3 lub większym, zapewniających dużą rozdzielczość obrazu np. Hewlett Packard lub Roland DG.

Komputer SICOMP PC umożliwia podłączenie interfejsu do sprzężenia z bardziej rozbudowanymi układami CAD pracującymi w ramach sieci komputerowej typu Ethernet, dzięki której następuje wymiana informacji i współpraca kilku jednostek. Software w opisywanej stacji graficznej stanowi system operacyjny GSX86 będący odpowiednikiem popularnego CP/M86 oraz oprogramowanie graficzne ZD CAD typu SIGRAPH.



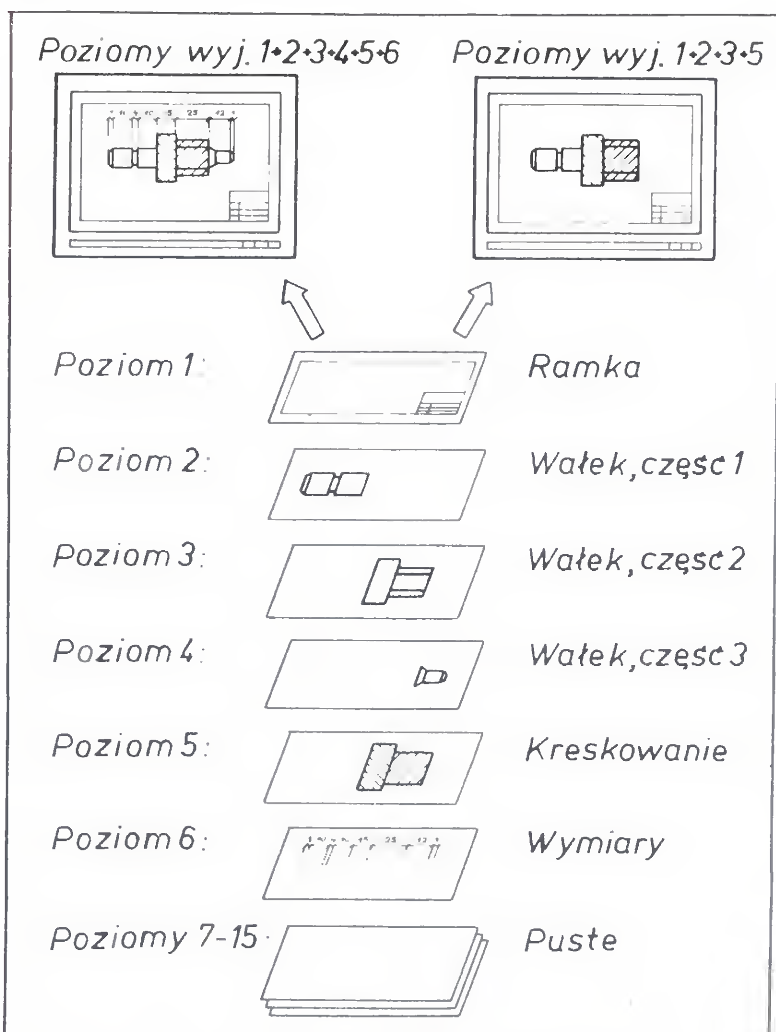
Schemat stacji graficznej WS10.



Wprowadzanie danych w systemie stacji graficznej WS10.

OBSŁUGA I DZIAŁANIE

System obsługuje użytkownik języka wejściowego albo rozgałęzionego „menu" ekranowego. Możliwości programowe SIGRAPH-u pozwalają na tworzenie rysunków z elementów podstawowych takich jak: punkt, prosta, okrąg, łuk okręgu, prostokąt.



Technika warstwowa tworzenia rysunków.

System posiada funkcje specjalne, do których należy wyznaczanie punktów przecięcia prostych, okręgu i prostej, rysowanie stycznych do okręgu itd.

Nanoszenie wymiarów podstawowych następuje automatycznie, zaś skomplikowane łańcuchy wymiarowe wprowadza użytkownik sam. W wygodny sposób dokonuje wszelkich opisów wybranym rodzajem liter (wielkość, pochylenie). Kreskowanie przekrojów może być standardowe lub można wprowadzić własne.

Dla usprawnienia pracy definiuje się symbole składające się z podstawowych elementów geometrycznych. Pod określoną nazwą gromadzone są one w pamięci i na życzenie mogą być odtwarzane. Pracownia może więc zestawić zbiór swoich własnych elementów podstawowych i oznaczeń.

Symbole mogą być dowolnie powtarzane na rysunku, wielokrotnie obracane, przesuwane, powiększane i pomniejszane, można tworzyć ich lustrzane odbicie, mogą być także wymiarowane lub oznakowane.

System wyposażony jest w tzw. technikę warstwową. Polega to na tym, że gotowy rysunek składa się z rysunków podstawowych na różnych poziomach.

Rysunki z różnych poziomów mogą być nakładane na siebie w dowolnej kombinacji jak przezroczyste kalki z naniesionymi na nich rysunkami składowymi. Można np. poziom 1 przeznaczyć na ramkę z tabliczką rysunkową poziomy 2, 3, 4 na rysunki części, poziom 5 na kreskowanie części, poziom 6 na opisy i wymiarowanie, a pozostałe poziomy będą niewykorzystane.

System SIGRAPH przewiduje 15 takich poziomów (warstw).

Na życzenie poszczególnych użytkowników firma SIEMENS dostarcza oprogramowanie CAD o konkretnym profilu np. konstrukcje mechaniczne, projektowanie zakładów przemysłowych, projektowanie obwodów elektrycznych. Pakiety te dysponują pełnym asortymentem symboli i oznaczeń oraz specjalistycznych programów obliczeniowych np. obliczenia wytrzymałościowe belek, dobór łożysk, kół zębatych (mechanika) do wykorzystania przez projektanta.

Technika CAD wypiera tradycyjną deskę kreślarską i suwak logarytmiczny, zapewniając bardziej komfortowy i wydajniejszy środek pracy konstruktora.

W wyniku dalszego rozwoju zastosowań CAD należy się liczyć z wyeliminowaniem z konkurencji tych ośrodków konstrukcyjnych, które nie dysponują tą techniką.

Grzegorz Kostrzyński
Andrzej Górecki

WARTO PRZECZYTAĆ



Małgorzata Kalinowska, Wacław Iszkowski „Klucze do BASICU". Warszawa 1987, Wydawnictwo Naukowo-Techniczne. Nakład 60 tys. egz. Cena 300 zł.

W serii „Mikrokomputery" publikowanej przez Wydawnictwa Naukowo-Techniczne ukazała się broszura „Klucze do Basicu". Autorzy tej książeczki, Małgorzata Kalinowska-Iszkowska i Wacław Iszkowski wpadli na bardzo oryginalny pomysł oddania czytelnikowi do rąk ściągawki ułatwiającej posługiwanie się różnymi wersjami BASIC-a. Przeznaczona jest ona głównie dla osób mających pewną wiedzę i doświadczenie, stąd sposób prezentowania informacji jest bardzo zwięzły. Omówiono w niej krótko podstawowe cechy charakterystyczne dla czterech popularnych u nas komputerów: Amstrad, Apple, IBM PC i ZX Spectrum. Oprócz tego znajdują się w niej listy poleceń występujących w BASIC-u każdego z tych komputerów wraz z opisem składni. Dla wygody pogrupowane są one tematycznie, np. sterowanie programem, grafika, generowanie dźwięku. Do każdej z list dołączono wykaz komunikatów błędów. Na końcu broszury zamieszczono ogólny spis słów kluczowych języka oraz krótki ich opis.

Książka stwarza okazję dokonania szybkiego porównania i oceny możliwości sprzętu, jest zwięzłym przeglądem różnych implementacji Basica. Może być cenną pomocą przy tłumaczeniu oprogramowania.

Szkoda, że autorzy nie rozszerzyli publikacji o wiadomości dotyczące Atari i Commodore. Użytkownicy tych dwóch grup komputerów na pewno będą żałować, że o nich zapomniano.

Sam pomysł wydania książki ściągawki warto zastosować i do innych zagadnień informatycznych. Dobrze opracowana pomoc tego typu bardzo ułatwiłaby życie ludziom stykającym się z najróżniejszym i niejednolitym pod względem rozwiązań hardware'owych i software'owych sprzętem. Może ktoś podjąłby się opracowania „Tablic informatycznych", które mogłyby zawierać np. listy rozkazów różnych mikroprocesorów, dokładne mapy pamięci komputerów itp.

(j.j.)

INFOKRAK '87

W dniach od 21 do 23 października odbyła się w Krynicy impreza o nazwie Krakowskie Spotkania Informatyczne INFOKRAK '87, poświęcona głównie możliwościom zastosowania informatyki w zarządzaniu przedsiębiorstwem.

Zainteresowanie tą problematyką dawno już wykroczyło poza granice środowiska zawodowego informatyków. Jest to zresztą logiczne — przecież komputeryzacja firmy jest problemem dyrekcji i całej załogi, a nie samych programistów. Znalazło to odbicie w składzie uczestników — byli wśród nich dyrektorzy, główni księgowi, ekonomiści. Byli także pracownicy naukowi z wyższych uczelni. Około połowy uczestników stanowili informatycy.

Organizatorami imprezy byli: Towarzystwo Naukowe Organizacji i Kierownictwa, Polskie Towarzystwo Informatyczne oraz Zakład Informatyki Akademii Ekonomicznej w Krakowie.

Czy w tak szerokim gronie można dyskutować o komputerach? O komputerach prawdopodobnie nie, ale celem spotkania była dyskusja o zastosowaniach, a ten temat musi być omawiany z udziałem wszystkich zainteresowanych.

Wymiana poglądów (niekiedy dość ostra) wykazała, że mamy trochę interesujących koncepcji i wartościowych wdrożeń, ale niestety nie mamy jeszcze w pełni sprawdzonych w praktyce, dobrze funkcjonujących wzorców, czy ogólnych metod postępowania.

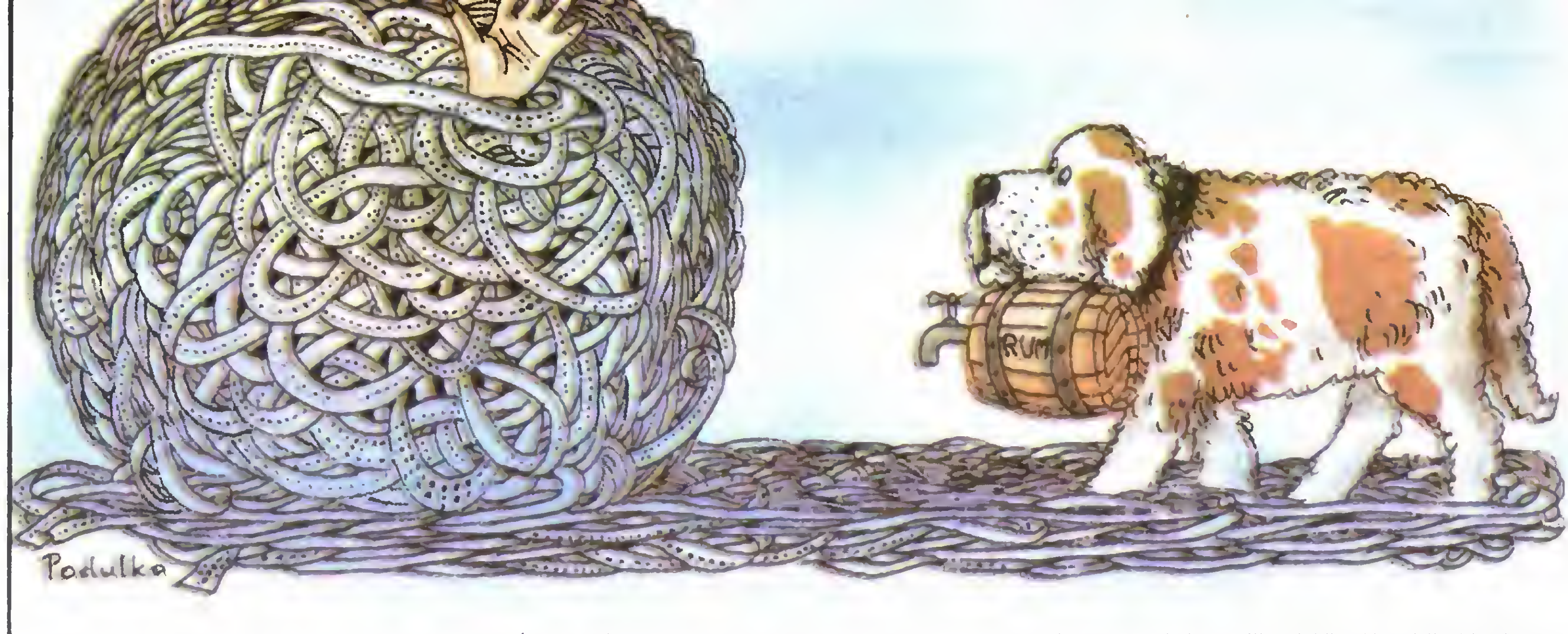
Moim zdaniem, na szczególną uwagę zasługuje to, że coraz częściej podkreśla się, iż jednym z ważnych czynników decydujących o pomyślności wdrożenia systemu informatycznego jest świadomość i psychika użytkownika. Jedna z czterech sekcji obrad zgromadziła referaty pod wspólnym tytułem: „Czynniki społeczne w zastosowaniach informatyki". Tak jak wielu autorów, uważam że w warunkach dobrego dostępu do sprzętu oraz niezłego oprogramowania narzędziowego, to właśnie czynniki społeczne będą stanowić najpoważniejszą barierę ograniczającą liczbę i zasięg udanych wdrożeń.

Oczywiście komputeryzować się nie można tylko teoretycznie, „na sucho", toteż uczestnicy konferencji z zainteresowaniem wzięli udział w „popołudniu z producentami", podczas którego swoje poglądy na dzień dzisiejszy polskiej informatyki, oraz swoje plany na przyszłość zaprezentowało kilka znanych krajowych firm produkujących sprzęt i oprogramowanie.

Sumując mogę stwierdzić, że takie imprezy są bardzo potrzebne i chętnie weźmę udział w następnym INFOKRAKU, obiecanym przez organizatorów przy pożegnaniu.

(ap)

NASTĘPNY KROK



SYSTEM OPERACYJNY

Dziś kontynuujemy przegląd podstawowych pojęć informatycznych.

System operacyjny, to program (lub raczej zespół programów) nadzorujący bezpośrednio pracę systemu komputerowego. Z programem tym użytkownik komputera spotyka się najczęściej — przy każdym kontakcie z maszyną. Myślę, że bliższe informacje na jego temat zainteresują zarówno użytkowników komputerów, jak też i ludzi obserwujących informatykę z boku.

Zaczniemy od historii, i to z kilku powodów. Funkcje, które dziś pełni system operacyjny pojawiały się sukcesywnie na przestrzeni lat — łatwiej będzie przedstawiać je kolejno, prezentując warunki w których powstały. Zobaczymy również, że rozwój oprogramowania był nierozdzielnie związany z rozwojem sprzętu. Łatwiej będzie zrozumieć to co się dzieje podczas rozwoju sprzętu mikro i systemów operacyjnych tego sprzętu.

Mam nadzieję, że taki powrót do historii pozwoli przypomnieć, że informatyka nie powstała wraz z mikrokomputerem. Wprost przeciwnie. Mikrokomputer jest jednym z jej elementów, powstał jako kolejny etap rozwoju. Zapewne wcale nie ostatni etap. I właśnie przyglądanie się historii może nasunie Wam podejrzenie, że biorąc pod uwagę realne proporcje, mikrokomputer wcale nie jest w informatyce najważniejszy — przynajmniej na razie.

Wróćmy jednak do naszych baranów, a konkretnie do tych zamierzchłych czasów, kiedy ogromnej większości Czytelników i Autorów „Bajtka" nie było jeszcze na świecie, za to istniały już i całkiem nieźle pracowały pierwsze elektroniczne maszyny liczące. Ich użytkownicy to nie przeciętni zjadacze chleba, lecz wysoko wyspecjalizowani fachowcy. Zwykle twórcy maszyny i autorzy programów oraz specjaliści oczekujący na wyniki obliczeń — ci ostatni byli niezbędni, bo ktoś przecież musiał sprawdzić, czy komputer dobrze policzył. Proces przygotowania obliczeń i ich nadzorowania był bardzo czasochłonny i wymagał głębokiej znajomości drutów maszyny oraz zagadnienia, które było rozwiązywane, a satysfakcja, że maszyna policzyła dobrze była chyba warta przynajmniej tyle co same wyniki obliczeń. Generalnie, na tym wstępnym etapie, człowiek był niezastąpiony w bezpośrednim kierowaniu pracą maszyny. Ci, którzy tutaj wykrzyknęli: to jest ten etap, na którym jestem ja i moje Spectrum lub Commodore, są niestety w błędzie, choćby dlatego, że wtedy nie było jeszcze BASIC-a. Gdyby jednak mieli rację, to nie zazdrościłbym im zbytnio — z naszego punktu widzenia posługiwanie się pierwszymi komputerami było piekielnie pracochłonne i skomplikowane. Jednak wtedy było to duże ułatwienie w pracy naukowców, którzy musieli wykonywać bardzo skomplikowane obliczenia numeryczne, więc człowiek i jego lampowy przyjaciel mogliby żyć długo i szczęśliwie, wspomagając się nawzajem, gdyby nie demon postępu, któremu zawsze mało tego co jest.

KOMPUTERY PRZYŚPIESZYŁY

Nowo powstające modele maszyn matematycznych liczyły coraz to szybciej. W takim razie musiały też szybciej wprowadzać dane i wyprowadzać wyniki obliczeń. Nie tylko jakość lecz i liczba komputerów rosła, więc można było rozwiązać coraz więcej zagadnień, lecz do tego celu trzeba było pisać coraz więcej programów. Ludziom znudziły się zera i jedynki, z których budowali programy. Pojawiają się pierwsze języki programowania, muszą się więc pojawić także translatory dla nich. Skutkiem tego proces wykonywania obliczeń wzbogaca się o liczne nowe elementy i oczywiście znacznie się komplikuje.

Przypomnijmy, że aby program mógł być wykonany przez procesor komputera, musi zostać zapisany w pamięci operacyjnej, w postaci ciągu rozkazów maszynowych (pisałem o tym w numerze 8/87). Tak więc programy przygotowane na nośnikach papierowych (taśmy lub karty perforowane), bez udziału komputera (ang. offline), muszą zostać przygotowane do wykonania. Najpierw musimy umieścić w pamięci gotową do wykonania treść programu translator, np. wczytując ją z taśmy perforowanej. Następnie uruchomić wykonanie tego programu. Uruchomiony translator musi wczytać treść programu do przetłumaczenia (taśma) i wygenerować kod maszynowy przetłumaczonego programu. Kod ten będziemy chcieli wykonać, więc powinien być umieszczany w pamięci komputera. A tam na razie siedzi pracujący translator, więc kod wynikowy będziemy na bieżąco perforować na taśmie, i wczytamy do maszyny gdy translator skończy pracę i będzie go można wyrzucić. Ostatecznie wczytany i uruchomiony program wynikowy zwykle potrzebuje danych (kolejna tasiemka!) i jeszcze trzeba mu przydzielić jakieś urządzenie do wyprowadzania wyników. Uff, straszne to, nawet do przeczytania a co dopiero do wykonania w praktyce. Szczególnie jeśli się ma świadomość, że za każdą minutę korzystania z komputera będę musiał słono zapłacić. (Milczeniem pomijam to co mógł mieć do powiedzenia programista, gdy przy samym końcu obliczeń program „padł" z powodu jakiegoś, nie wykrytego wcześniej, błędu).

ZAPLĄTANY TAŚMAMI

Programista — użytkownik nie był chyba zbyt szczęśliwy, tym bardziej, że oprócz tego musiał jeszcze być zapatrzony w lampki. Zwróćmy uwagę, że za beztroskim stwierdzeniem: „trzeba wczytać taśmę" kryje się sporo pracy. Wczytanie taśmy przez maszynę, to przecież też wykonanie przez procesor serii rozkazów. Skąd biorą się w pamięci te rozkazy — przecież nie z taśmy, bo to błędne koło. Musimy mieć do dyspozycji tak zwany pulpit techniczny, z którego (naciskając guziczki) będziemy się uprzejmie zwracać do procesora, i na którym (oglądając zapalające się i gasnące lampki) będziemy czytać informacje o stanie maszyny. Oczywiście komunikacja w obie strony wyłącznie w języku maszyny!

Widać już chyba wyraźnie dwie poważne rzeczy do zrobienia — należy zgromadzić procedury odpowiadające typowym, wykonywanym przez operatora czynnościom, np. załadowanie programu binarnego z taśmy do pamięci, uruchomienie wykonania programu, itd.

— trzeba zacząć kontaktować się z maszyną bez użycia pulpitu technicznego. W tym celu dołączono do komputera elektryczną maszynę do pisania. Ponieważ jednak procesor zupełnie nie rozumiał pisanych na niej poleceń, trzeba jeszcze napisać program, który będzie wczytywał pisane przez operatora teksty, interpretował je i wykonywał. Jak wykonywał? Oczywiście przez wywołanie (uruchomienie) odpowiedniej procedury. Np. napisanie polecenia LOAD powoduje wczytanie z taśmy programu binarnego i umieszczenie go w pamięci; RUN — rozpoczęcie wykonywania obliczeń, itd.

To co powstanie, to program nadzorczy, przodek naszego systemu operacyjnego. Żeby nasze rozwiązanie miało wartość praktyczną program ten musi cały czas siedzieć w pamięci operacyjnej. Jeśli polecimy wykonać program użytkownika, to sterowanie powinno być przekazane do tego programu, a ostatnią instrukcją użytkownika powinno być przekazanie sterowania do nadzorcy, który wczyta następne polecenie operatora, wydane z... monitora — tak właśnie nazywała się ta elektryczna maszyna do kontaktu z komputerem. Dziś to znaczenie słowa „monitor" zanika.

Zwróćmy uwagę na daleko idące skutki tego rozwiązania (uproszczenie korzystania z komputera było celem i jest oczywiste, ale są także inne skutki — niby uboczne, ale o ogromnym znaczeniu)

POJAWIA SIĘ NOWA JAKOŚĆ

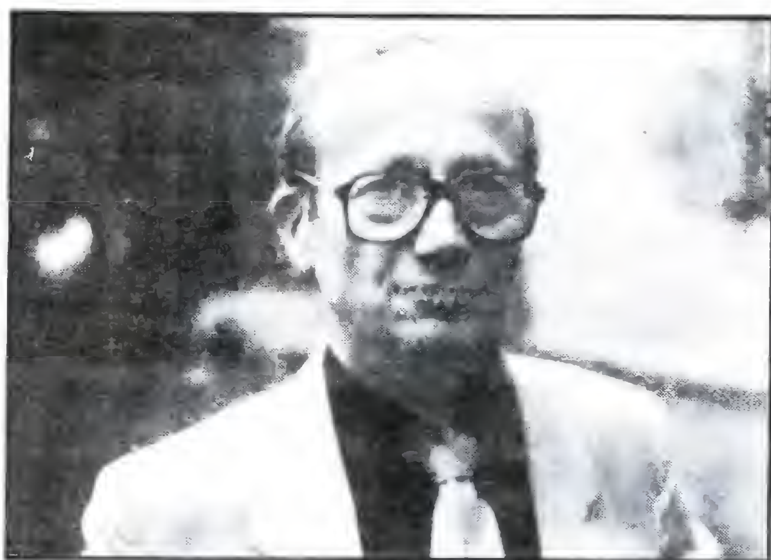
Powstał program ułatwiający współpracę człowieka z maszyną. Jest to zwykły program, musi być zapisany w pamięci, musi być wykonywany przez procesor. Skutkiem tego jest podział zasobów komputera między programy użytkowe i systemowe — część pamięci i czasu procesora x) trzeba na stałe przeznaczyć dla programu nadzorczego. Nie jest to przejściowa dolegliwość — w przyszłości będzie się to pogłębiać. Dojdzie nawet do tego, że konstruktorzy nowych maszyn będą musieli uwzględniać w swoich projektach specyficzne potrzeby systemów operacyjnych. Taka jest cena za zwalenie na komputer nadzoru nad procesem przetwarzania naszych zadań.

Skoro w systemie operacyjnym znajdują się procedury obsługi urządzeń wejścia/wyjścia, to pisząc program nie muszę już bezpośrednio oprogramowywać każdej swojej transmisji, (sprawdzać, czy czytnik gotów, inicjować czytania znaku, sprawdzać, czy wczytany poprawnie, itd) — mogę skorzystać z procedury systemowej. To z kolei powoduje powiązanie programu już nie tylko z konkretną maszyną (sprzętem), lecz i z konkretnym systemem operacyjnym. Program działający pod kontrolą jednego systemu operacyjnego może nie zadziałać na tym samym komputerze pod kontrolą innego systemu. Czyli sam sprzęt przestał być jednoznacznym określeniem możliwości. Należy raczej zacząć mówić o systemie komputerowym, mając na myśli nie tylko sprzęt lecz i oprogramowanie systemowe, oba te elementy tworzące uzupełniającą się całość. Dzisiejszy użytkownik często nie rozróżnia funkcji wykonywanych bezpośrednio przez sprzęt, od funkcji wykonywanych przez oprogramowanie. W codziennej praktyce nie jest to wcale konieczne. Warto sobie jednak uświadomić, że bez oprogramowania systemowego komputer (oczywiście także mikrokomputer) to kupka złomu o niewielkiej wartości użytkowej. Warto również, przynajmniej z grubsza, znać funkcje i rozumieć działanie tego oprogramowania, dlatego w przyszłym miesiącu systemów operacyjnych ciąg dalszy.

(cdn)

Andrzej Pilaszek

x) Szczególnie dotkliwe było to w przypadku pamięci operacyjnej, gdyż tego właśnie zasobu zwykle było za mało.



Rozmowa z inż. Ivanem Malcem — doradcą ds. komputerowych ministra przemysłu elektrotechnicznego CSRS.

JESTEM SCEPTYKIEM

— Ile mikrokomputerów użytkuje się obecnie w Czechosłowacji?

— W przybliżeniu 70 tysięcy sztuk. Z tego około 20-25 tys. naszej produkcji. Reszta pochodzi z NRD, gdzie kupiliśmy ok. 17 tys. sztuk i z państw kapitalistycznych.

— W Tusexie dostać można minikomputery Atari i Spectrum, w agencji prasowej instaluje swoje komputery Siemens, drukarnia bratysławskiej „Prawdy" pracuje w oparciu o zachodnioniemiecki Linotype-6,... A gdzie młody Czechosłowak kupić może za korony swój mikrokomputer?

— Praktycznie nie może.

— A przecież jeszcze w ubiegłym roku...

— Zgadza się. Nasz handel za nadwyżkę dewiz zakupił ok. 2 tysiące mikrokomputerów, które sprzedawano w zależności od typu w cenie 5-10 tys. koron.

— Jakie to były mikrokomputery?

— Atari — w większości — i Sinclair Spectrum.

— Dlaczego zrezygnowano z tej formy sprzedaży?

— Nie wiem czy zrezygnowano, jeśli w tym roku handel zarobi więcej dewiz to być może przeznaczy je na zakup kolejnej partii mikrokomputerów. Nie chcąc powiększać długu przeznaczyliśmy na zakup mikrokomputerów tylko nadwyżkę dewiz. Jestem być może nieobiektywny w swoich ocenach, ale uważam, iż jako społeczeństwo nie dorośliśmy jeszcze do powszechnego wykorzystywania mikrokomputerów — wyłączając oczywiście gry. Znacznie poważniejszy jest problem wyposażenia w nie szkół, obecnie we wszystkich szkołach średnich wykorzystywane jest ok. 7 tys. komputerów — 5-10 w każdej szkole.

— Jak zatem wytłumaczy pan absolwentowi takiej szkoły, który wie już, jak i po co korzystać z komputera, że nie może go sobie kupić?

— Są przecież kółka mikrokomputerowe...

— Siedem klubów Atari, niewiele więcej Spectrum.

— Będzie więcej. Również z myślą o prywatnym rynku obniżyliśmy opłaty celne za przywóz minikomputerów z zagranicy.

— Stwierdził pan, że jest być może nieobiektywny w swoich ocenach. Ilu jeszcze minister musi mieć subiektywnych doradców, żeby podjąć obiektywną decyzję?

— Od spraw komputerowych jestem jedynym doradcą. Nie znaczy to jednak, że minister decyduje kierując się moimi opiniami. Wielu ludzi, cały aparat ministerstwa przygotowuje optymalne rozwiązanie problemu.

— O tym, iż należy stawiać na elektronikę i komputery przekonywać pana nie trzeba. Jak pan przekonuje oponentów?

— Jak już powiedziałem, nie jestem sam.

— Jednak w tym m.in. celu, popularyzacji, wyjaśniania i przekonywania, stworzył pan nowe pismo „Elektronika". Czy nakład 80 tys. jest wystarczającym argumentem?

— Mógłby być większy, ale przydział papieru jest limitowany.

— Kto czyta „Elektronikę"?

— Rozchodzimy się praktycznie bez zwrotów, pomimo że, w porównaniu z innymi czasopismami na naszym rynku jesteśmy o ok. 30-40 proc. drożsi. Jesteśmy popularni, choć nie wszyscy zgadzają się z koncepcją pisma. Konstruktorzy — amatorzy narzekają, że poświęcamy ich problemom za mało miejsca; użytkownicy komputerów, mając do nas pretensje, że za mało w piśmie programów, stawiają nam za przykład Bajtka.

— Mogą go sobie bez problemu zaprenumerować!

— Dla nas nie jest to wyjście, ale oczywiście, że mogą. Sam przeglądam go regularnie.

— Podobno już w styczniu przyszłego roku otwarty będzie w Pradze pierwszy w Czechosłowacji „dom software'owy"?

— Niestety, nie będzie. Walczę od lat o powołanie centrum programowego, opracowującego koncepcje i sposoby — można powiedzieć: narzędzia — ułatwiające tworzenie dobrych programów aplikacyjnych.

— W przypadku programów, zwłaszcza edukacyjnych, sprawa wydaje się być o tyle istotna, iż kiepskie narobią więcej szkody niż przyniosą pożytku.

— Obecnie najważniejszym jest — według mnie — niedopuszczenie do monopolu instytutów naukowych jako jedynych twórców programów oraz braku konkurencji między nimi w tej dziedzinie.

— Jakie są ceny i kto płaci programiście za stworzenie oryginalnego programu?

— Użytkownik programu może dogadać się z programistą i zapłacić mu cenę umowną.

— Na przykład?

— Możliwość taka istnieje dopiero od miesiąca.

— Czy pana zdaniem, programy powinno się zabezpieczać?

— Sądzę, iż programy powinny być zabezpieczane przez programistę i formalnie na zasadzie praw autorskich. Uważam, że programy będą lepsze, bardziej oryginalne, jeśli wiadomo będzie konkretnie, kto jest ich autorem.

— Luka, przepaść, która dzieli nas w tej dziedzinie od czołówki światowej wydaje się nie do przeskoczenia. Czy ma pan algorytm wyjścia z tego technologicznego dołka?

— Jestem sceptykiem.

Rozmawiał
Franciszek Penczek

LEKCJE BEZ STOPNI

Austriacy mają już za sobą dyskusje na temat tego, czy i jak wprowadzać informatykę do szkół. Przedmiot pod nazwą elektroniczne przetwarzanie danych zaczęto wprowadzać do szkół w 1970 r. Najpierw był to przedmiot do wyboru, a od 1977 r. stał się obowiązkowym składnikiem programu nauczania we wszystkich szkołach technicznych. Jednakże prawdziwą rewolucję przeżyło austriackie szkolnictwo w ubiegłym roku, kiedy informatykę jako przedmiot obowiązkowy wprowadzono do szkół ogólnokształcących.

Wypracowanie programu nauczania tego przedmiotu było dziełem specjalnej komisji, w skład której wchodzili przedstawiciele Ministerstwa Wychowania, Kultury i Sportu, związków zawodowych, pracodawców, pedagodzy, ekonomiści oraz inni fachowcy. W rekordowo krótkim, rocznym okresie ustalono nie tylko sam program nauczania, ale również sposób kształcenia nauczycieli, kierunki zakupu sprzętu, potrzeby finansowe oraz społeczne następstwa wprowadzenia nowego przedmiotu. Informatyka została wprowadzona w ilości 2 godzin tygodniowo do klas 9-tych, natomiast w klasach 10-12 można ją wybrać jako przedmiot dodatkowy.

Problem braku wykładowców przedmiotu rozwiązano w ciągu roku organizując 2 tygodniowe kursy dla nauczycieli przedmiotów ścisłych. Uznano, że w pierwszym etapie 80 godzin wykładów zapewni wystarczający poziom nauczania. Co roku podwyższano niezbędny poziom wyszkolenia nauczycieli, który obecnie wynosi minimum 5 do 8 tygodni kursu.

W 1985 r. zakupiono sprzęt — 1000 komputerów osobistych firm Honey Bull, Philips i Toshiba kompatybilnych z IBM PC. Okazało się, że nawet dla Austriaków oryginalne IBM-ki są za drogie. Przeciętna szkoła wyposażona jest w sześć PC. Obecnie prowadzone są negocjacje w sprawie zakupu nowej partii 1000 maszyn, co w rezultacie powinno doprowadzić do otrzymania 8 stanowisk pracy w każdym szkolnym laboratorium.

Szczególny nacisk jest kładziony na zintegrowanie nauczania informatyki z innymi przedmiotami. Chodzi przede wszystkim o to, aby w ramach przedmiotu nauczyć wykorzystywania komputera w różnych dziedzinach pracy i życia.

Największym problemem pozostaje oczywiście oprogramowanie. Opracowanie, przetestowanie i wdrożenie programu trwa od roku do 2 lat i kosztuje ok. 300 tys. szylingów. W celu pozyskania rodzimych programów rozpisano konkurs wśród nauczycieli, którego efekty będą znane w końcu tego roku.

Aby nie zrażać uczniów do nowych technologii wypracowano dość ciekawy system oceniania. Przez pierwsze pół roku nie stawia się stopni. Po pierwszym semestrze wystawiane jest świadectwo z ogólną oceną postępów i wtedy uczeń musi się zdecydować, czy na świadectwie końcowym chce mieć wystawiony stopień czy nie. W praktyce okazuje się, że 25 do 30% słuchaczy nowego przedmiotu nie życzy sobie wpisywania ocen do świadectwa. Niski procent rezygnujących z klasyfikacji nie dziwi, wziąwszy pod uwagę, że w coraz większej liczbie zawodów znajomość informatyki jest niezbędna do otrzymania pracy.

Marek Rudziński

WSZYSTKO DLA WSZYSTKICH

JAK REKLAMOWAĆ SIĘ W BAJTKU?

Reklamy przyjmuje Młodzieżowa Agencja Wydawnicza (Redakcja Wydawnictw Poradniczych i Reklamy). 04-028 Warszawa. Al. Stanów Zjednoczonych 53, pokój 313, tel. 10-56-82.

Cena reklamy biało-czarnej wynosi 300 zł za 1 cm^2. Do ceny podstawowej doliczane jest 30% za dodatkowy kolor i 100% w przypadku reklamy wielobarwnej. Ogłoszenie drobne kosztuje 200 zł za jedno słowo.

SPRZĘŻENIE ZWROTNE

Drogi Bajtku!

Piszę niejako w uzupełnieniu odpowiedzi udzielonej w „Bajtku" 9/87 koledze Sławomirowi Jóźwiakowi w sprawie równoczesnego wykonywania programów BASIC-owych i programu muzycznego. Znam dwie możliwości rozwiązania tego problemu. Pierwszą, o której pisałeś — czyli wykorzystanie przerwań, oraz druga — niejako „na okrętkę" — wykonanie z taśmy magnetofonowej z nagranym podkładem muzycznym oraz sygnałami synchronizacyjnymi sterującymi wykonywaniem programu głównego. Potrzebny jest wtedy również króciutki podprogram odczytujący w/w sygnały. Jeżeli chodzi o sposób pierwszy, to jest on bez wątpienia bardziej „elegancki" i wygodniejszy. Poza tym wcale nie ma konieczności samodzielnego pisania programu w języku maszynowym. Istnieje dość rozpowszechniony na naszym rynku oprogramowania program SOUNDMACHINE, który oprócz umożliwienia wygodnego zapisu muzyki z wykorzystaniem wszystkich możliwości brzmieniowych komputera ATARI pozwala wygenerować (opcją CREATE) gotowy program maszynowy odtwarzający z wykorzystaniem przerwania VBI zapisany utwór muzyczny równocześnie z wykonywaniem dowolnego programu w języku BASIC. Jedynym problemem pozostaje napisanie programu wczytującego i uruchamiającego wygenerowany plik maszynowy. W wersji dyskowej dyskietka z programem SOUNDMACHINE zawiera także program demonstrujący możliwość „udźwiękowienia" programów w BASIC-u, oraz program AUTOMAT.BAS, który takie udźwiękowienie umożliwia. Niestety, sposób działania tego programu (tworzy on plik typu AUTORUN.SYS możliwy do wykorzystania tylko przy pracy ze stacją dysków) uniemożliwia współpracę z pamięcią kasetową.

Przyjmując nieco inną zasadę działania napisałem kasetową wersję programu łączącego i chętnie ją udostępnię zainteresowanym osobom.

Jeśli chodzi o metodę drugą, dosyć dobrym przykładem jej wykorzystania jest krążący program pt. „6 lekcji języka ATARI-BASIC", w którym na taśmie, oprócz zasadniczego programu, nagrany jest tekst czytany przez lektora oraz sygnały sterujące wykonaniem programu głównego.

Swoimi doświadczeniami na ten temat chętnie podzielę się ze wszystkimi zainteresowanymi.

Marek Górski
ul. Sukiennicza 3 m. 120
91-851 Łódź

Dziękuję za cenne uzupełnienie danej przeze mnie odpowiedzi. Sam nie zetknąłem się uprzednio z taką możliwością programu SOUNDMACHINE, stąd brak wzmianki o nim.

Produkcja mikroprocesorów rozpoczęła się od produkcji 4- i 8-bitowych. Obecnie w powszechnych zastosowaniach profesjonalnych królują procesory 16-bitowe, aczkolwiek jest już w użyciu wiele maszyn wyposażonych w jednostki 32-bitowe.

Logiczną ekstrapolacją tego procesu byłyby mikroprocesory 64-bitowe. W związku z tym nasuwa mi się pytanie: czy są prowadzone prace w kierunku stworzenia mikroprocesorów 64-bitowych oraz czy wprowadzenie ich do profesjonalnych komputerów osobistych wniosłoby tylko zmiany ilościowe (większa pamięć operacyjna i szybkość) czy również jakościowe?

Robert Repucha
ul. Świrkowa 11
19-500 Gołdap

Prace tego typu z pewnością są prowadzone, chociaż na razie trudno jest w prasie zagranicznej odnaleźć miarodajne informacje na ten temat — takie badania z reguły objęte są tajemnicą. Zadanie zaprojektowania i wyprodukowania mikroprocesora 64-bitowego jest jednak z pewnością bardzo trudne i pociąga za sobą szereg problemów technicznych, które w obecnym stanie nie są jeszcze rozwiązane. Takie problemy wiążą się również z samym procesem projektowania, który wymaga użycia komputerów wielkiej mocy przetwarzania.

Trudno określić, jakie dokładnie zmiany jakościowe (a na pewno takie będą) pociągnie za sobą wejście na rynek procesorów 64-bitowych. Warto jedynie zauważyć, że długość słowa maszynowego nie ma bezpośredniego wpływu na wielkość przestrzeni adresowej, wyznaczanej szerokością szyny adresowej procesora.

Nasuwa się także spostrzeżenie, że w chwili obecnej nie ma wyraźnego zapotrzebowania na popularny mikrokomputer z procesorem 64-bitowym. Rozwój oprogramowania mikrokomputerów wyraźnie nie nadąża bowiem za rozwojem sprzętu. Wielodostęp, równoległe przetwarzanie — oto możliwości stworzone przez zastosowanie procesorów 32-bitowych, które jednak rzadko są wykorzystywane — ze względu na brak odpowiedniego oprogramowania.

Jestem posiadaczem TIMEX-a 2048. Mam często potrzebę pisania tekstów i używania grafiki (32 kolumny i 256 x 175 to trochę mało), a wiem, że można w tym komputerze otrzymać większą rozdzielczość przez OUT 255,6.

Po wykonaniu powyższego OUT otrzymujemy zwężone litery, ale nadal w 32 kolumnach i standardową grafikę. Jak korzystać z tego trybu?

Jarosław Musialik
ul. Pawia 37/40
Lublin, woj. legnickie

Po wykonaniu instrukcji OUT 255,6 pamięć obrazu zostaje przeorganizowana w następujący sposób: dotychczasowa, standardowa pamięć ekranu odpowiada zawartości parzystych kolumn obrazu (a zatem 0,2,4,...62). Zawartość kolumn o nieparzystych numerach umieszczona jest w tzw. pamięci ekranu 2, to znaczy w bloku zaczynającego się od adresu 24567 i sięgającego do 31487.

Używana z poziomu Basic-a instrukcja PRINT działa tylko na pamięci obrazu 1. Stąd właśnie w trybie wysokiej rozdzielczości znaki są węższe, ale też ukazują się w co drugiej kolumnie ekranu.

Pełne wykorzystanie możliwości grafiki wysokiej rozdzielczości na komputerze Timex wymaga osobnego zaimplementowania procedur grafiki i wypisywania tekstu dla tego trybu. Da się tego dokonać i z poziomu Basic-a, takie działanie jest jednak pozbawione sensu ze względu na zbyt małą szybkość. Radziłbym w tym celu skopiować, odpowiednio modyfikując, procedury systemowe PRINT_ FP (2DE3), PRINT_ STRING (203C). Utworzenie procedur graficznych byłoby trudniejsze i w zasadzie polegać tu trzeba na własnej pomysłowości. Warto jednak przyjrzeć się przedtem systemowym podprogramom PLOT_ SUB (22E5), PIXEL_ ADD (22AA), DRAW_ 3 (24BA).

Liczby w nawiasach stanowią szesnastkowe adresy podanych procedur.

Marcin Waligórski

	GIEŁDA „BAJTKA" (tys. zł)	PEWEX BALTONA (USD)	Sklep „Bajtka"
SINCLAIR			
ZX 81	25–30	—	30–35
ZX Spectrum 48 KB	70–85	115	100–110
ZX Spectrum Plus	110–120	—	140–150
ZX Spectrum 128 + 2	200	—	—
Drukarka SEIKOSHA GP 50S	65–85	—	110–120
TIMEX 2048	95–100	146	—
Joystick	4,5–7	—	8–12
COMMODORE			
C-64	160–170	219	220–230
C-128	240	299	340–360
C-128D	690	—	850–900
Amiga z monitorem kolorowym	1.2 mln	—	—
Magnetofon 1531	35	48	48
Stacja dyskietek 1541	180	—	210–220
Stacja dyskietek 1571	240	299	270–290
Drukarka MPS-803	180	—	200–220
Dyskietki 5 1/4" (średnia jakość)	0.65–1.5	3.50	0.8–1
ATARI			
65XE	105–110	125	130–140
1300 XE	165	199	—
Stacja dyskietek 1050	180–190	187	210–220
Drukarka 1029	170–180	199	230–250
ATARI 520 STM st.dysk. SF-314	900	998	—
AMSTRAD			
464 z monit. monochromat.	210	—	280–300
6128 z monit. monochromat.	390	—	650–700
6128 z monitorem kolorowym	450	—	—
Dyskietki 3"	3–5	—	4–6
Stacja dyskietek 3" do 464	290	—	—
PC 1512	1,2–1,3	—	—

Sklep „Bajtka" znajduje się w Bytomiu przy ul. Jainty 19

PPC 512 S

Nowością, niestety jeszcze nie krajowej giełdy jest nowy typ Amstrada. Pierwszy przenośny mikrokomputer tej firmy, o którym w przeciwieństwie do poprzednich modeli dowiadujemy się na 2—3 miesiące przed zaplanowanymi dostawami do sklepów.

PPC 512 S jest właściwie odpowiednikiem CPC 1512, posiada 10 calowy monitor plazmowy z 80 kolumnami zasilanie bateryjne, kartę graficzną kompatybilną z MDA i CGA o rozdzielczości 640x200 i waży tylko 5.5 kg. Cena tego mikrokomputera ma oscylować w granicach 500 dolarów amerykańskich, czyli o połowę taniej od zbliżonych mikrokomputerów Toshiba i Compaq.

Zainteresowanych odsyłamy do firmy Polanglia, która specjalizuje się w pośrednictwie i sprzedaży wszystkich Amstradów. Tel. z Polski 0-0441-840 1715



Tomasz Kryszeń, uczeń Szkoły Podstawowej, 14 lat. Posiada Atari 65 XE i magnetofon XC 12. Zainteresowania: sklejanie modeli, informatyka. Oprogramowanie: gry oraz programy użytkowe. Nawiąże kontakt w sprawie wymiany informacji o grach i innych programach. Adres: 80-169 Gdańsk, ul. Kamińskiego 5/182.

Marcin Primus, lat 14. Posiada minikomputer PHILIPS VG 8020 system MSX, monitor, magnetofon i stację dysków. Zainteresowania: matematyka, informatyka, chemia i fizyka. Oprogramowanie gry, programy edukacyjne. Proponuje wymianę oprogramowania. Adres: 43-100 Tychy ul. Gorkiego 40/15.

Monika Duda, uczennica lat 8. Posiada Commodore 16 oraz Amstrad (Schneider 6128). Oprogramowanie: (napisane przez tatę) rachunki dla dzieci w wieku 5—12 lat, geometria, ćwiczenie wyrazów i pisowni oraz wiele programów firmowych. Proponuje wymianę programów i literatury oraz wymianę doświadczeń. Adres: D-6750 Kaiserslautern, Barbarossaring 32, RFN.

Jaromir Bajkowski, uczeń lat 15. Posiada komputer Amstrad-Schneider 464. Zainteresowania: wędkarstwo, literatura SF. Adres: 31-057 Kraków, ul. Piekarska 11/12.

Marek Górski, student Politechniki Łódzkiej lat 28. Posiada ATARI 65 XE, magnetofon, monitor Neptun. Oprogramowanie: programy muzyczne, graficzne, języki. Zainteresowania: informatyka, muzyka, elektronika. Podejmie korespondencję w celu wymiany oprogramowania, literatury i doświadczeń w pracy z ATARI. Adres: 91-851 Łódź, ul. Sukiennicza 3/120.

Mariusz Wyszyński, technik elektronik, 24 lata. Mikrokomputer ATARI 800 XL, stacja dysków 1050. Oprogramowanie: programy użytkowe, gry. Proponuje wymianę programów i doświadczeń. Adres: 96-515 Teresin ul. Wąska 1.

Zygmunt Siwiński, lat 34 technik mechanik. Mikrokomputer ATARI 800 XL, magnetofon XC 12. Oprogramowanie: około dwustu gier i programów użytkowych. Proponuje wymianę oprogramowania z użytkownikami Atari. Adres: 95-035 Ozorków ul. Zachodnia 13 m 15.

Pavel Kośik, posiada komputer ZX SPECTRUM. Nawiąże korespondencję w celu wymiany czasopism komputerowych, literatury fachowej oraz doświadczeń. Korespondencja w języku czeskim, adres: MIŠKOVECKA 16, 040II KOŠICE, CSSR.

Ulf Gebhard, student lat 23. Zainteresowania: literatura, komputery, informatyka, gry komputerowe. Nawiąże korespondencje w języku niemieckim w celu wymiany doświadczeń. Adres: R. ROTHKEGEL-STR. 47, DDR-7500 COTTBUS.

Leszek Spychała, student Politechniki. Posiada minikomputer ATARI 520 STMF. Oprogramowanie: gry, programy demonstracyjne. Zainteresowania: informatyka oraz muzyka. Nawiąże korespondencję w celu wymiany doświadczeń. Adres: 46-200 KLUCZBORK, ul. Krakowska 4.

Marcin Knapik, lat 14, minikomputer SHARP MZ-721. Proponuje wymianę firmowych gier oraz programów użytkowych. Adres: 82-200 MALBORK, ul. Dzierżyńskiego 10/B/4



PROGRAMY KOMPUTEROWE KAW

Krajowa Agencja Wydawnicza w Warszawie podjęła we wrześniu 1986 r. profesjonalną produkcję kaset z programami komputerowymi.

Wszystkie programy oferowane są w trzech wersjach: ZX SPECTRUM, ATARI 800 XL/65XE/130XE, COMMODORE 64. Dotychczas ukazały się następujące pozycje:

- **BIORYTMY/NUMERY SZCZĘŚCIA**

Programy dla zwolenników wiedzy ezoterycznej. Skupiają przy monitorach całe rodziny: wnuczek sprawdza swój biorytm przed randką lub treningiem, babcia nie musi już chodzić do wróżki-kabalistki.

- **TIXO/NIM-2**

Mało znane gry logiczne dla wszystkich, których znudził już kosmiczny rozbój i laserowy terror. Ze względu na swe walory dydaktyczne mogą być wykorzystane w szkole, jako ciekawy temat zajęć dodatkowych z matematyki.

- **KÓŁKO I KRZYŻYK**

Gra wszystkim dobrze znana, jednak zawsze emocjonująca. Po raz pierwszy realizacja komputerowa wersji pięciopolowej! Twój komputer nie da Ci w tej grze żadnych szans!

- **HEXAN**

Gra dla miłośników labiryntów. O powodzeniu nie decyduje sprawność w niszczeniu klawiatury lub joysticka, lecz umiejętność rozwiązywania różnego rodzaju problemów. Z grą związany jest konkurs — po dotarciu do SKARBCA możesz brać udział w losowaniu nagród!

- **FUNKCJA KWADRATOWA**

Program edukacyjny dostosowany jest do wymagań polskiego programu nauczania, oferuje różnorodne formy prezentacji i sprawdzania wiadomości.

- **RACHUNEK PRAWDOPODOBIEŃSTWA cz. I**

„Przyjazny" program i kompetentny, cierpliwy nauczyciel. Niestandardowe ujęcie tematu oraz ciekawa grafika to duże atuty tego programu. W przygotowaniu dalsze pozycje z tej serii.

- **BASIC TEST**

Czy znasz dobrze BASIC swojego komputera? Jeśli nie, to program ten pomoże Ci poznać go lepiej. Jeśli myślisz, że tak, okaże się pewnie, że znajdzie się coś, czego możesz się jeszcze nauczyć. Doskonała pozycja dla tych, którym nie wystarcza już używanie swojego komputera jedynie jako zabawki.

(szar)

ŻAGLÓWKA I KOMPUTER

Hołny Mejera to malownicza miejscowość w północno-wschodniej Polsce. Niewielkie jezioro, plaża, kawałek lasu. Nieco wyżej na skarpie odrestaurowany dworek oraz wiekowy park. Cisza i spokój. Wymarzone miejsce wypoczynku, ale nie tylko. Bo oto w dwóch salach dworku, który okazuje się być Ośrodkiem Szkoleniowo-Wypoczynkowym Politechniki Białostockiej znalazły się z początkiem września monitory, komputery i napędy dyskowe, a także drukarki i plotery, połączone jak należy kablami i przedłużaczami, a poza tym pudła z dyskietkami, czyli wszystko to co trzeba by rozpoczął pracę obóz mikrokomputerowy organizowany już po raz trzeci przez białostocką uczelnię.

Wprawdzie zdaniem organizatora mgr. Krzysztofa Iwaszko tych 6 IBM-ów, 3 Schneidery 6128 oraz Atari 65XE i 130XE, to jednak zbyt mało, gdyż na tym etapie umiejętności konieczny jest już całkowicie indywidualny dostęp do komputera no i nie przez godzinę czy nawet dwie. A trzeba powiedzieć, że w pokojach Ośrodka zamieszkało 22 uczestników, pragnących przez dwa tygodnie dzielić czas pomiędzy klepaniem w klawiaturę a pływaniem żaglówką.

W programie obozu znalazły się m.in.: Turbo Pascal, edytory, Wordstar, Chiwriter oraz Lettrix, program graficzny Fantasy i baza danych dBase III. Były również gry, a obozową listę przebojów wygrał Boulderdash IV w wersji na Atari.

Andrzej Karwowski

Cześć Maluchy!

Po raz pierwszy w naszej rubryce zajmiemy się programowaniem w LOGO. Wielu Przedszkolaków probowało już oswajać sympatycznego żółwia LOGO, wielu chciałoby się nauczyć z nim rozmawiać. Wiem o tym z Waszych listów. Spróbujemy więc dzisiaj zaprzyjaźnić się z tym miłym zwierzakiem.



OSWAJAMY ŻÓŁWIA

Wersje LOGO dla poszczególnych komputerów różnią się nieznacznie między sobą. Istnieją także polskie wersje tego języka dla Atari i Spectrum. My będziemy korzystali z wersji angielskiej, dostępnej dla większości mikrokomputerów.

Słyszeliście zapewne, że programowanie w LOGO to zupełnie co innego niż programowanie w BASIC-u i to nie tylko dlatego, że w BASIC-u nie ma żółwia. Spróbujmy przekonać się, na czym ta inność polega. Rozpoczynamy więc rozmowę z komputerem. Na ekranie pojawił się znak zapytania

?

oznacza to, że komputer czeka na polecenia. Napiszmy więc:

? st

co oznacza „pokaż żółwia". I rzeczywiście, po naciśnięciu ENTER/RETURN na ekranie ukazuje się żółw w kształcie trójkąta. Napiszmy teraz:

? fd 100

czyli „naprzód" o 100 kroków — żółw przesuwa się przed siebie a jego ogonek rysuje kreskę. Piszemy dalej:

? rt 90

„w prawo" o 90 stopni — żółw skręca posłusznie i ustawia się pod kątem prostym do narysowanej kreski.

Już widzę Wasze zawiedzione miny — przecież bardzo podobnie postępuje się w BASIC-u, co w tym niezwykłego? Proszę jednak o trochę cierpliwości. Teraz wytłumaczymy komputerowi, lub jeśli wolicie, żółwiowi, jak się rysuje kwadrat.

```
?to kwadrat
› fd 100 rt 90
› fd 100 rt 90
› fd 100 rt 90
› fd 100 rt 90
›end
```

Jak zauważyliście, uczenie żółwia rozpoczynamy od słowa **to** (w polskich wersjach **oto**) a kończymy słowem **end** (w polskiej wersji **juz**). Pomiędzy tymi słowami podaje się czynności, które ma wykonać żółw. Wystarczy teraz napisać

? kwadrat

i żółw rysuje żądaną figurę.

Ucząc komputer nowych słów możemy korzystać z poprzednich, np.:

```
?to dwakwadraty
› kwadrat
› rt 180
› kwadrat
›end
```

A to już zupełnie co innego. Prawda? Zwróćcie uwagę, że słowo „dwakwadraty" napisane jest łącznie. Słowa w LOGO nie mogą zawierać spacji.

Wiemy już wystarczająco dużo, by napisać prostą grę. Jest to gra dla dwóch osób, które na zmianę sterują żółwiem. Kto wyprowadzi żółwia poza dozwolony obszar lub „nadepnie" na pozostawiony ślad, ten przegrywa. Żółw może się przesuwać prosto, w prawo lub w lewo o tę samą odległość. Dla ułatwienia słowa sterujące żółwiem będą jednoliterowe.

n — prosto
p — prawo
l — lewo

Dobrze byłoby, gdyby te objaśnienia pojawiały się podczas gry. Nauczmy więc komputer wypisywać je.

```
?to napis
› ct
› pr [n — prosto p — prawo l — lewo]
› end
```

ct oznacza kasowanie wszystkich tekstów.
pr to polecenie drukowania.

Teraz nauczymy żółwia wykonywać poszczególne komendy.

```
?to p
› rt 90
› fd 30
› napis
›end
?to l
› lt 90
› fd 30
› napis
›end
?to n
› fd 30
› napis
›end
```

W następnej kolejności wytłumaczymy żółwiowi jak rysuje się ramkę.

```
?to ramka
› cs pu
› fd 150 rt 90 fd 240 pd
› repeat 2 [rt 90 fd 240 rt 90 fd 480]
› pu home pd
›end
```

cs to rozkaz czyszczenia ekranu.
pu oznacza „podnieś pisak" — od tej chwili aż do otrzymania polecenia **pd** „opuść pisak" żółw nie będzie zostawiał śladu na ekranie.
repeat 2 jest to polecenie powtórzenia dwa razy czynności umieszczonych w nawiasach kwadratowych.
home powoduje powrót żółwia do pozycji startowej.

Na koniec nauczymy komputer rozpoczęcia gry.

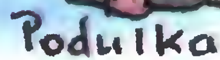
```
?to gra
› ramka
› napis
›end
```

Zwróćmy uwagę na jeszcze jedną odmienność programu napisanego w LOGO i BASIC-u. Grając w naszą grę wydajemy komputerowi przez cały czas polecenia w LOGO, podczas gdy w BASIC-u musielibyśmy stosować instrukcje INPUT lub INKEY$ i odpowiednie skoki. LOGO pozwala więc na znaczne uproszczenie programu.

Sądzę, że wszyscy polubiliście żółwia LOGO. Jeśli tak, to będziecie mieli okazję spotykać się z nim częściej.

Romek

KONKURS ŚWIĄTECZNY



1. Na gwiazdkę dostanę joystick i dyskietki lub dostanę łyżwy.
2. Na gwiazdkę dostanę joystick i dostanę dyskietki lub łyżwy.
3. Na gwiazdkę dostanę joystick lub dyskietki i dostanę łyżwy.
4. Na gwiazdkę dostanę joystick lub dostanę dyskietki i łyżwy.
5. Na gwizdkę dostanę joystick lub dostanę dyskietki lub dostanę łyżwy.

	A	B	C	D	E
1					
2					
3					
4					
5					

tabela I

TABELA ZMIENNYCH

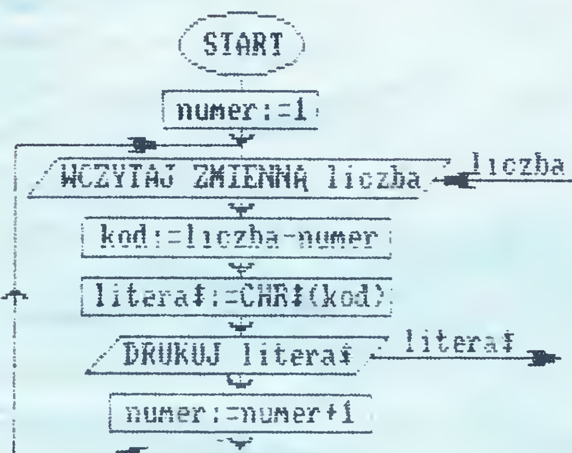
numer	liczba	kod	litera$	drukarka

tabela II

Kubuś Literka z wielką niecierpliwością oczekiwał Świąt. Szczególnie interesowało go jakie gwiazdkowe prezenty znajdzie tym razem pod choinką. Próbując przewidzieć zamiary Świętego Mikołaja zapisał sobie pięć różnych zdań. Pod choinką znalazło się pięć worków z prezentami dla Kubusia i jego rodzeństwa. Nie wiemy który przeznaczony był dla naszego przyjaciela. Ustalmy natomiast, czy kolejne zdania zapisane przez Kubę byłyby prawdziwe, gdyby otrzymał on poszczególne worki z prezentami. Wyniki wpiszemy do tabelki I. Jeśli na przykład zdanie nr 2 jest prawdziwe w przypadku otrzymania worka C, w odpowiedniej rubryce wpisujemy, 1, jeśli jest fałszywe piszemy O.

Po wypełnieniu tabeli możemy przystąpić do pokonywania łańcucha na naszej choince logicznej. Wędrówkę rozpoczynamy od szczytu. Odczytujemy kolejne zera i jedynki z tabeli I (poziomo) i przemieszczamy się wzdłóż łańcuchów, pomiędzy zawieszonymi na gałązkach bombkami. Jedynka oznacza, że powinniśmy wybrać łańcuch zielony, zero — łańcuch czerwony. Oczywiście po drodze zapisujemy starannie liczby umieszczone na bombkach.

Otrzymany ciąg cyfr jest szyfrem, który trzeba zdekodować przy pomocy algorytmu przedstawionego w postaci schematu blokowego.



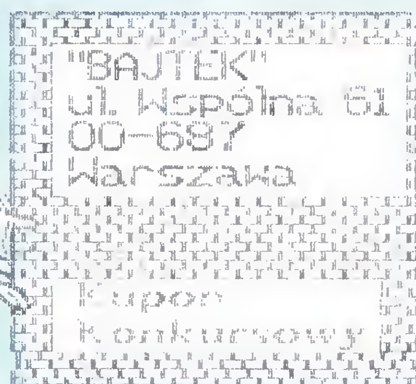
Można to zrobić przy pomocy komputera lub... ołówka. W tym ostatnim przypadku z pewnością przyda Wam się tabela zmiennych (tabela II), do której wpisuje się kolejne wartości zmiennych występujących w programie. Przykład takiego dekodowania dla liczb: 76, 119, 101, 101 przedstwiony jest poniżej.

TABELA ZMIENNYCH

numer	liczba	kod	litera$	drukarka
1	76	75	K	Kuba
2	119	117	u	
3	101	98	b	
4	101	97	a	

Rozwiązania wraz z kuponem konkursowym należy przysyłać do 15 lutego 1988 roku.

Na tych, którzy nadeślą prawidłowe rozwiązania czeka komputer Atari 65 XE z magnetofonem ufundowany przez PZ KAREN oraz inne atrakcyjne nagrody. *(b)*



32	
33	!
34	"
35	#
36	$
37	%
38	&
39	'
40	(
41	)
42	*
43	+
44	,
45	-
46	.
47	/
48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7
56	8
57	9
58	:
59	;
60	<
61	=
62	>
63	?
64	@
65	A
66	B
67	C
68	D
69	E
70	F
71	G
72	H
73	I
74	J
75	K
76	L
77	M
78	N
79	O
80	P
81	Q
82	R
83	S
84	T
85	U
86	V
87	W
88	X
89	Y
90	Z
91	[
92	\
93	]
94	^
95	_
96	`
97	a
98	b
99	c
100	d
101	e
102	f
103	g
104	h
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
117	u
118	v
119	w
120	x
121	y
122	z
123	{
124		
125	}
126	~

NIE TYLKO KOMPUTERY

MARS PO „FOBOSIE"

dokończenie ze str. 32

Ekspedycje na Marsa trzeba wysyłać w momencie, kiedy znajduje się on z przeciwnej strony Słońca niż Ziemia. Do końca bieżącego stulecia takie

okna astronomiczne

do startów na Marsa będą pojawiały się mniej więcej co 2 lata. Uwzględniając te terminy uczeni radzieccy przedłożyli do dyskusji swoim zagranicznym kolegom program kolejnych etapów badań Marsa. Końcowym celem tego programu będzie dostarczenie na Ziemię — do 2000 roku — gruntu tej planety.

Misja „Fobos" może być rozpatrywana jako pierwszy bardzo ważny, krok w realizowaniu tego programu. Następny etap przewiduje budowę sztucznego satelity Marsa, którego zadaniem będzie obfotografowanie całej planety i wykonanie badań w atmosferze i na powierzchni Marsa.

Jeśli pojazd kosmiczny wystartuje z Ziemi we wrześniu 1992 roku, to po 340 dobach w sierpniu 1993 roku osiągnie on okolice Marsa. Po wejściu satelity na orbitę okołomarsjańską z jego pokładu zostanie wystrzelone urządzenie posiadające specjalną aparaturę napędową służącą do zejścia z orbity. Na wysokości około 13 kilometrów włączy się system spadochronów, następnie zostaną zrzucone na powierzchnię aparaty analizujące fizykochemiczne właściwości gruntu, a do atmosfery planety wprowadzony zostaje aerostatyczny balonsonda.

Przeprowadzono wstępną analizę zestawu aparatury naukowej przeznaczonej dla każdego elementu ekspedycji. Powinien się tam znaleźć system kamer telewizyjnych mogących filmować z odległości od 5 do 10 metrów, spektrostrefowy polarymetr do badań makroreliefu powierzchni, spektometr gamma do badania składu chemicznego poszczególnych jej gatunków, zestaw plazmowy, magnetometr. Aby wybrać najbardziej korzystne miejsca do lądowania dla przyszłych ekspedycji potrzebne są dokładne i długotrwałe obserwacje tych samych rejonów powierzchni Marsa. W tym celu trzeba umieścić na platformie obrotowej kamery telewizyjne, dzięki czemu możliwe będzie automatyczne ich naprowadzanie i utrzymanie w odpowiednim kierunku.

Pomiary w podczerwieni pozwolą na wykonanie termicznej mapy powierzchni. Kompleks aparatury meteorologicznej planuje się umieścić w gondoli balonu.

„Marsochody" — naprzód

Dane otrzymane w trakcie tej ekspedycji pozwolą na przygotowanie kolejnego etapu badań — dostarczenia na Marsa „marsochodów". Na tym etapie użyteczniejszy będzie inny schemat pojazdu kosmicznego, przewodujący rozdzielenie go na dwa bloki: orbitalno-retranslacyjny i lądujący z samobieżnym „marsochodem" na pokładzie.

Aby „marsochód" (tzn. pojazd poruszający się po powierzchni planety) mógł swobodnie zmieniać swoje położenia, trzeba wyposażyć go w określone „intelektualne" możliwości. Pojazd taki powinien np. umieć omijać przeszkody, których 20–30 minut wcześniej nie spotkał na swej drodze — przykładowo tyle czasu potrzeba, aby sygnały radiowe dotarły z Marsa na Ziemię i z powrotem. Dlatego też, oprócz systemu telewizyjnego, „marsochód" trzeba będzie wyposażyć w laserowy dalmierz korygujący pojazd. Zasięg poruszania się „marsochodu" powinien być obliczony na setki kilometrów. Prędkość ruchu pojazdu będzie uzależniona od zasobu energii a także od rzeźby terenu i programu badawczego na trasie jego poruszania się. W charakterze źródła energii, zasilającej pojazd, mogą być wykorzystane albo baterie słoneczne albo termoelektrogeneratory izotopowe.

Operacja przechwycenia

Dostarczenie marsjańskich próbek gruntu na Ziemię wydaje się być najbardziej skomplikowanym elementem przedłożonego przez radzieckich uczonych programu. Możliwy jest następujący wariant: zostaną wystrzelone dwa autonomiczne pojazdy kosmiczne. Jeden z nich wyląduje na powierzchni Marsa a drugi będzie jego satelitą na orbicie okołoplanetarnej. Pojazd, lądujący powinien mieć na pokładzie rakietę nośną i niewielki pojazd samobieżny, który pobierze próbki gruntu w pewnej odległości od miejsca lądowania.

Rakieta nośna, która wystartuje z pojazdu znajdującego się na powierzchni Marsa, połączy się z pojazdem krążącym na orbicie. Tam marsjańskie próbki zostaną przeładowane do specjalnego pojazdu wracającego na Ziemię. Kiedy znajdzie się on w pobliżu naszej planety — zostanie przechwycony przez okołoziemską stację orbitalną.

Najprawdopodobniej wstępna analiza przywiezionych próbek wykonana zostanie na pokładzie stacji orbitalnej. Pozwoli to na rozwiązanie jednego z najtrudniejszych problemów całej ekspedycji — problemu kwarantanny. Chodzi o wykluczenie możliwości zarażenia naszej planety organizmami pozaziemskimi. Trzeba o tym pomyśleć choćby sama taka możliwość była trudna do uwierzenia. Rozumie się też samo przez się, że konieczna jest sterylizacja pojazdu kosmicznego przed startem z Ziemi, aby nie zarazić z kolei Marsa ziemskimi mikrobami.

Operacja połączenia na orbicie wokołomarsjańskiej została już w zadowalającym stopniu opracowana przez specjalistów radzieckich. Jednakże, biorąc pod uwagę limit ciężaru podczas lotów międzyplanetarnych, rysuje się jeszcze ogromna praca w dziedzinie skonstruowania bardzo lekkich systemów połączeniowych i agregatów.

Kolejny etap badań powierzchni Marsa mógłby rozpocząć się w latach 2000–2005 przy pomocy dużych „marsochodów" z wydłużonym okresem działania i zasięgiem do tysiąca kilometrów.

Następnie, około roku 2010, wysłana zostałaby

ekspedycja kombinowana

pozwalająca na lądowanie dwóch–trzech pojazdów samobieżnych na Marsie i pobieranie próbek z dwóch-trzech różnych miejsc.

W wyniku tego cyklu badań około 2020 roku zostaną stworzone warunki dla ekspedycji pilotowanej — z lądowaniem kosmonautów na Marsie.

Przewiduje się, że proponowanym przez uczonych radzieckich programie badawczym Marsa wezmą udział organizacje naukowe oraz specjaliści z wielu innych krajów. Doświadczenie związane z realizacją programu „Wega" wykazało jak efektywna może być taka kooperacja.

Lot człowieka na Marsa jest niemożliwy bez współpracy międzynarodowej, zjednoczenia wysiłków wielu krajów. Planeta, która otrzymała imię starożytnego boga wojny może dzięki takiemu lotowi stać się symbolem jednoczenia narodów w pokojowym opanowaniu i wykorzystaniu przestrzeni kosmicznej.

Jurij Zajcew



MARS PO „FOBOSIE”

LOT CZŁOWIEKA NA MARSA JEST NIEMOŻLIWY BEZ WSPÓŁPRACY MIĘDZYNARODOWEJ



Czy człowiek poleci na Marsa? Amerykański uczony Carl Sagan uważa, że istnieją już wszystkie przesłanki do wykonania takiego lotu. Niedawno odbył się telemost specjalistów radzieckich i amerykańskich, podczas którego omawiano techniczne i polityczne aspekty takiego przedsięwzięcia. Być może już wkrótce dowiemy się o konkretnych planach w tym zakresie...

Oczywiście, lot człowieka na Marsa musi być częścią szerokiego programu badań tej planety. Poprosiliśmy więc Jurija Zajcewa, kierownika wydziału w Instytucie Badań Kosmicznych Akademii Nauk ZSRR, aby zapoznał czytelników „Bajtka” z radzieckimi planami badań Marsa, w okresie pomiędzy rozpoczynającą się w przyszłym roku misją „Fobos” (o której pisaliśmy w „Bajtku” nr ...) a ewentualnym lotem człowieka na Czerwoną Planetę.

Doświadczenie minionych lat wykazało, że najbardziej efektywny z naukowego punktu widzenia kosmiczny program badawczy to kilka lotów ku jednemu obiektowi z kilkuletnią przerwą. Chodzi o to, aby każda ekspedycja tworzyła bazę pod nowe, bardziej złożone eksperymenty. Tak właśnie wyglądał radziecki program badawczy Księżyca i Wenus. Podobnie będzie również wyglądał radziecki program badań Marsa.

Dlaczego Mars?

Ma on bardzo interesującą przeszłość. W szczególności, nie wyklucza się istnienia niegdyś na tej planecie otwartych zbiorników wodnych i gęstej atmosfery: na fotografiach wykonanych z pokładu pojazdów kosmicznych wyraźnie widoczne są jakby owiane wiatrem wzgórza i doliny, suche koryta rzek. Mars wciąż jeszcze jest jednym z nielicznych zakątków Układu Słonecznego, wobec którego ma się nadzieję, chociaż bardzo niewielką, na znalezienie jakichś form życia czy śladów jego istnienia w przeszłości.

Jest to pierwsza planeta, na której możliwe jest lądowanie kosmonautów. Lot ten jest wyjątkowo kuszący. Zresztą już sam fakt obecności człowieka na pokładzie lecącego ku Czerwonej Planecie statku kosmicznego czyni go interesującym, jako że interesujący jest sam człowiek, jego życie i praca w kosmosie.

Zdaniem większości uczonych start pilotowanej ekspedycji na Marsa powinien być poprzedzony wystrzelaniem w jego kierunku serii automatycznych pojazdów kosmicznych, które pozwoliłyby, w kolejnych etapach, na wypracowanie techniki lotów na tę planetę. Pozwoliłoby to również na wcześniejszy wybór najbardziej interesujących rejonów planety dla kolejnych lądowań i przeprowadzenie tam niezbędnych poszukiwań.

dokończenie na str. 31